



Technische Universität Ilmenau

Fakultät für Informatik und Automatisierung

Fachgebiet Neuroinformatik und Kognitive Robotik

Autonome Situationserkennung im klinischen Umfeld

Masterarbeit zur Erlangung des akademischen Grades Master of Science

Thanh Quang Trinh

Betreuer: Dipl.-Inf. Erik Einhorn

Dipl.-Inf. Christoph Weinrich

Dipl.-Inf. Steffen Müller

Verantwortlicher Hochschullehrer:

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

Die Masterarbeit wurde am 23.09.2014 bei der Fakultät für Informatik und Automatisierung der Technischen Universität Ilmenau eingereicht.

[urn:nbn:de:gbv:ilm1-2015210265](https://nbn-resolving.org/urn:nbn:de:gbv:ilm1-2015210265)

Erklärung: „Hiermit versichere ich, dass ich diese Masterarbeit selbständig verfasst und nur die angegebenen Quellen und Hilfsmittel verwendet habe. Alle von mir aus anderen Veröffentlichungen übernommenen Passagen sind als solche gekennzeichnet.“

Ilmenau, 23.09.2014

.....
Thanh Quang Trinh

Zusammenfassung

Im Rahmen des Forschungsprojektes ROREAS (Interaktiver robotischer Reha-Assistent für das Lauf- und Orientierungstraining von Patienten nach Schlaganfällen) soll ein robotischer Laufassistent für eine Klinikumgebung entwickelt werden. Der Roboter soll Schlaganfallpatienten bei ihrem eigenständigen Lauftraining unterstützend begleiten. Das Lauftraining wird hauptsächlich auf den Korridoren der Klinik ausgeführt. Auf diesen Korridoren können Engstellen auftreten. Wenn der Roboter und eine Person sich gleichzeitig durch eine Engstelle bewegen, kann dies zu Verklemmungssituationen führen. Zur Gewährleistung einer höflichen und nutzerzentrierten Navigation, die ein wichtiger Bestandteil für die soziale Akzeptanz des Roboters ist, müssen Verklemmungssituationen im Voraus erkannt werden, um rechtzeitig eine Behandlung einzuleiten. Ziel der vorliegenden Masterarbeit ist die Entwicklung eines Verfahrens zur vorausschauenden Erkennung von Verklemmungssituationen. Zur Erkennung von Verklemmungen werden Engstellensituationen durch reellwertige Merkmalsvektoren beschrieben. Diese erfassen die Platzkonflikte des Roboters mit einer Person in Bezug auf erkannte Engstellen. Als Teil der Merkmale wird die Bewegung von Personen prädiziert, um Platzkonflikte vorherzusagen, die möglicherweise zu Verklemmungen führen können. Indem die Merkmalsvektoren zu Klassen zusammengefasst werden, welche die Behandlung ihrer entsprechenden Situation repräsentiert, kann die Erkennung von Verklemmungssituationen als Klassifikationsproblem betrachtet werden. Folglich kann ein Klassifikator eingesetzt werden, um die Situationen direkt auf ihre Behandlung abzubilden. Als Klassifikatoren wurden in dieser Arbeit eine lineare Support Vektor Maschine und ein manuell designter Entscheidungsbaum genutzt. Die experimentellen Untersuchungen dieser Arbeit zeigen, dass der gewählte Merkmalsraum geeignet ist, um Verklemmungssituationen zu erkennen. Der Entscheidungsbaum erzielte dabei auf den Datensätzen dieser Arbeit eine bessere Erkennungsrate als die lineare Support Vektor Maschine.

Abstract

The goal of the research project ROREAS (Robotic Rehabilitation Assistant for Stroke Patients) is the development of a robotic rehabilitation assistant for the self-training of stroke patients. The self-training aims at improving the walking and orientation skill of the patient. It consists mainly of a goal oriented movement in an environment of a rehabilitation center. The self-training is mostly performed on the hallways connecting the patient's rooms. Due to the structure of the building or objects staying in the hallways the lateral space is limited forming narrow passages. Moving in such a confined space imposes deadlocks in narrow passages. Since a polite and attentive navigation is an important requirement for an assistive robot, these deadlock situations must be recognized in advance to trigger a proactive reaction. In this master thesis an approach is presented for anticipating deadlock situations caused by narrow passages. In a nutshell, situations concerning narrow passages are captured by real-valued feature vectors. Basically, the features capture the structure of the environment along the robot's movement in terms of a possible narrow passage and the possible space conflicts caused by a person. As part of the features, the movement of the persons are predicted allowing the robot to forecast possible space conflicts resulting in the considered problematic situations. By grouping the feature vectors into classes representing the appropriate treatments of their corresponding situation, the recognition task becomes a classification problem. Thus, a classifier which solve this classification problem can be used to map each feature vector which represents a situation directly to the appropriate treatment. A linear support vector machine and a handcrafted decision tree is used as a classifier. The experimental evaluations show that the used features are good suitable for recognizing deadlock situations. The decision tree performed better on the dataset of this thesis as the linear support vector machine.

Inhaltsverzeichnis

1	Einleitung	1
1.1	ROREAS	1
1.2	Aufgabenstellung	2
1.3	Gliederung	3
2	Überblick und Grundlagen	5
2.1	Entwickelter Ansatz	6
2.2	Formulierung des Lösungsansatzes	7
2.3	Einordnung in das Softwaresystem von ROREAS	9
2.4	Aufbau des Situation Classifiers	11
2.5	State of the Art	12
3	Merkmalsextraktion	15
3.1	Lineare Prädiktion von Platzkonflikten	15
3.1.1	Merkmale	15
3.1.2	Bestimmung der Trajektorien	16
3.1.3	Bestimmung der Kollisionen	19
3.2	Engstellenerkennung	22
3.2.1	Merkmale	22
3.2.2	Berechnung der Durchfahrtsbreite	24
3.2.3	Trajektorie mit Bogenlängenparametrisierung	25
4	Klassifikation	29
4.1	Einleitung	29
4.2	Support Vektor Maschine	30

4.3	Entscheidungsbaum	32
5	Experimentelle Untersuchungen	35
5.1	Datensatz	35
5.1.1	Aufnahmeverfahren	35
5.1.2	Aufnahmeumgebung	36
5.2	Modeltraining	37
5.3	Auswertung	38
6	Zusammenfassung und Ausblick	45
6.1	Zusammenfassung	45
6.2	Ausblick	47
6.2.1	Datenerfassung	47
6.2.2	Patientenbezogene Merkmale	48
6.2.3	Prädiktive Kollisionserkennung	48
6.2.4	Berücksichtigung der Personenbewegung in der Engstellenerkennung	52
6.2.5	Weitere Ansätze	53
6.2.6	Parkposition	54
A	Entscheidungsbaum	57
A.1	Entscheidungsbedingungen	57
A.1.1	Bewegungsintention	58
A.1.2	Relative Lage der Person zum Roboter	59
A.1.3	Relative Lage zur Engstelle	59
A.1.4	Reaktionsbereich	61
A.1.5	Kritischer Kollisionsbereich	63
A.1.6	Vermeidung von unnötigen Reaktionen	64
A.2	Stehende Person	64
A.3	Mitbewegung	65
A.4	Gegenbewegung	66
A.5	Tabelle der Parameter	67
B	Ergänzende Unterlagen	73
B.1	Konfusionsmatrix	73

B.2 Quellenangabe	74
Abbildungsverzeichnis	75
Literaturverzeichnis	77

Kapitel 1

Einleitung

Roboter gewinnen zunehmend mehr als autonome mobile Assistenzsysteme an Bedeutung. So wurden sie schon in Einkaufsmärkten als Shopping-Guides oder in häuslichen Bereichen zur Unterstützung von Senioren eingesetzt. Mit dem demographischen Wandel entwickelt sich ein neuer Trend des Einsatzes von Robotern im Rehabilitationsbereich. Für diese Anwendungsszenarien ist das Navigationssystem eine unerlässliche Komponente der Roboterarchitektur, die eine zielgerichtete Bewegung in einer dynamischen Umgebungen ermöglicht. In einer solchen dynamischen Umgebung können komplexe Situationen auftreten, die vom Roboter erkannt und behandelt werden müssen, um eine sichere und höfliche nutzerzentrierte Navigation zu gewährleisten.

1.1 ROREAS

Die vorliegende Arbeit wird im Rahmen des Projektes *ROREAS* ausgeführt. ROREAS steht für *Interaktiver robotischer Reha-Assistent für das Lauf- und Orientierungstraining von Patienten nach Schlaganfällen* und ist ein derzeit laufendes Verbundprojekt des Fachgebietes für *Neuroinformatik und Kognitive Robotik* (NIKR). Neben den Fachgebiet gehört die Fachklinik Bad Liebenstein von der *m&i-Klinikgruppe Enzensberg*¹ zu den Projektpartnern.

Bewegungstraining ist ein wichtiger Bestandteil der Schlaganfall-Nachsorge. Ein regelmäßiges Training hilft nicht nur die physischen Fähigkeiten zu verbessern, sondern auch die kognitiven Fähigkeiten. Das Training wird meist nur unter Koordination eines Therapeuten in regelmäßigen Kursen durchgeführt. Ein eigenständiges Training außerhalb der Kurse fin-

¹m&i-Klinikgruppe Enzensberg: <http://www.enzensberg.de>



Abbildung 1.1: Eine Luftaufnahme der Fachklinik Bad Liebenstein [GROSS et al., 2014]. Der Einsatzort des robotischen Laufassistenten ist die Fachklinik Bad. Dort soll der Roboter den Patienten beim Lauftraining unterstützend begleiten.

det häufig nicht statt, obwohl diese zu einer schnelleren Rehabilitation führen kann. In der Fachklinik Bad Liebenstein wird daher jedem Patienten empfohlen, zusätzlich zu den verschriebenen Training, ein eigenständiges Lauftraining auszuführen. Das Lauftraining besteht in der zielgerichteten Fortbewegung im Klinikgebäude, und soll im Wesentlichen die Mobilität und das räumliche Orientierungsvermögen stärken. Allerdings kommt es oft vor, dass Schlaganfallpatienten Schwierigkeiten haben sich an die neue Umgebung der Klinik zu gewöhnen, sodass der Patient durch ein Krankenpfleger begleitet werden muss. Für die Krankenpfleger bedeutet dies eine zusätzliche Arbeitsbelastung.

Im Projekt soll deshalb ein robotischer Reha-Assistenz entwickelt und in der Fachklinik Bad Liebenstein integriert werden, der den Patienten beim eigenständigen Lauftraining unterstützend begleitet und das Training auf abrechenbarer Weise dokumentiert. Dadurch soll das noch kaum genutzte Verbesserungspotential des Eigentrainings stärker in die Rehabilitation integriert werden und gleichzeitig die Krankenpfleger entlasten.

1.2 Aufgabenstellung

Der konkrete Einsatzort des robotischen Assistenten ist die Fachklinik Bad Liebenstein (siehe Abbildung 1.1). Die Klinik ist ein Gebäudekomplex bestehend aus einem mehrstöckigen Hauptgebäude, in den die Patienten untergebracht sind und weiteren Anbauten für Behandlungszwecke. Der Roboter soll vorrangig im Innenbereich des Hauptgebäudes eingesetzt werden. Das eigenständige Lauftraining wird dort hauptsächlich auf den Korridoren ausgeführt

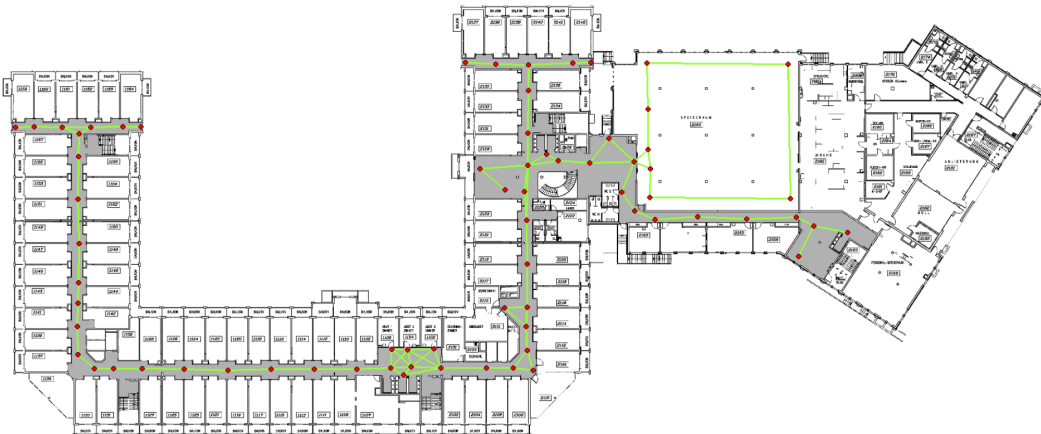


Abbildung 1.2: Grundriss einer Etage der Fachklinik Bad Liebenstein [GROSS et al., 2014]. Das Lauftraining findet auf den Korridoren der Fachklinik statt. In den Grundriss sind mögliche Wege dieses Training dargestellt.

und besteht aus einer zielgerichteten Fortbewegung im Klinikgebäude. In Abbildung 1.2 ist der Grundriss der Fachklinik mit möglichen Laufwegen dargestellt.

Aufgrund der Gebäudestruktur ist der seitliche Platz in einigen Bereichen der Korridore sehr stark begrenzt, sodass Engstellen entstehen können. Zudem werden die Korridore vom Klinikpersonal zum Abstellen von Arbeitsgegenständen (z.B. Betten, Rollstühle, Reinigungswagen) genutzt. Dadurch entstehen zusätzliche Engstellen, die allein aus dem Grundriss nicht zu erkennen sind. Der seitliche Platz in den Engstellen ist so begrenzt, dass keine zwei Personen nebeneinander Platz haben. Wenn der Roboter und eine Person sich gleichzeitig durch die Engstelle bewegen möchten, kann dies zu Verklemmungssituationen führen.

Ziel dieser Arbeit ist die Entwicklung eines Verfahren zur Erkennung von solchen Verklemmungssituationen. Dabei ist es wichtig diese Situationen schon im Voraus zu erkennen, um proaktiv eine Behandlung einzuleiten. Dies ist ein wichtiger Bestandteil für eine höfliche und nutzerzentrierte Navigation. Zudem soll das Verfahren in das Robotersystem von ROREAS integriert und anschließend evaluiert werden.

1.3 Gliederung

Im folgenden Kapitel 2 wird zuerst der entwickelte Ansatz in einem Überblick zusammengefasst. Darauf aufbauend wird der Ansatz formal beschrieben. Zudem wird die softwaretechnische Einordnung in das Robotersystem von ROREAS erläutern und der Forschungsstand zu

Verfahren der expliziten Situationserkennung für die Roboternavigation vorgestellt. Das entwickelte Verfahren nutzt Merkmale, um die Verklemmungssituationen zu beschreiben. Diese werden im Kapitel 3 vorgestellt. Ausgehend von den Merkmalen wird ein Klassifikator genutzt, um Situationen voneinander zu unterscheiden. Die verwendeten Klassifikatoren werden im Kapitel 4 erläutert. In Kapitel 5 werden die experimentellen Untersuchungen zur Bestimmung der Performance des Verfahren beschrieben. Mit einer Zusammenfassung und einen Ausblick wird die Arbeit im Kapitel 6 abgeschlossen.

Kapitel 2

Überblick und Grundlagen

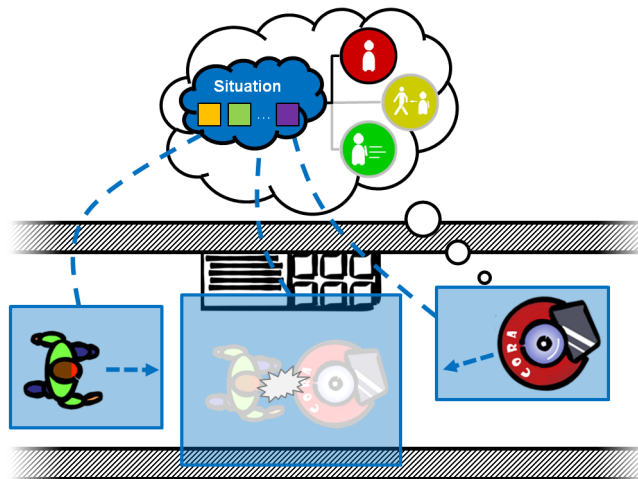
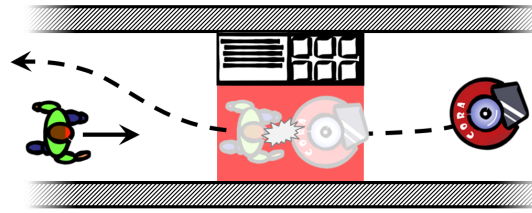


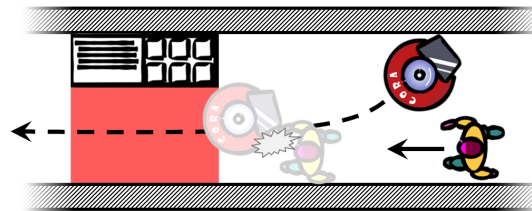
Abbildung 2.1: Schematische Darstellung des merkmalsbasierten Klassifikationsansatzes zur Erkennung von Verklemmungssituationen. Jede Situation wird auf ein Merkmalsvektor reduziert, das die Engstelle und die möglichen Platzkonflikte mit den Personen entlang der Roboterbewegung beschreibt. Die Merkmalsvektoren werden zu Klassen zusammengefasst, welche die Behandlung ihrer Situation entsprechen. Ein Klassifikator kann folglich zur direkten Abbildung von Situationen auf deren Behandlung genutzt werden.

In den folgenden Abschnitten dieses Kapitel wird der entwickelte Ansatz in einem Überblick vorgestellt. Darauf aufbauend wird dieser Lösungsansatz formal beschrieben und das realisierende Softwaremodul mit der Einordnung in das Robotersystem von ROREAS vorgestellt. Anschließend wird der Stand der Forschung zur expliziten Situationserkennung für die Roboternavigation vorgestellt.

2.1 Entwickelter Ansatz



(a) Parksituation mit einer entgegenkommenden Person



(b) Einreihungssituation mit einer mitlaufenden Person

Abbildung 2.2: Schematische Beispiele der betrachteten Klassen von Verklemmungssituationen.

Das Ziel von ROREAS ist die Entwicklung eines robotischen Assistenten für eine Klinikumgebung, der Schlaganfallpatienten bei ihren eigenständigen Lauftraining auf den Korridoren unterstützend begleiten soll. Auf den Korridoren können Engstellen entstehen, in denen der zeitliche Platz nur für eine Person oder den Roboter ausreicht. Eine solche Engstelle trennt den Korridor in zwei Bereiche ab. Um von einem Bereich in den Nächsten zu gelangen, gibt es keinen effizienteren Weg als durch die Engstelle. Die Engstelle gibt somit ein Bewegungsfluss vor, der gleichzeitig eine Bewegung nur in einer Richtung erlaubt. Wenn der Roboter und eine Person sich gleichzeitig durch die Engstelle bewegen möchten, kann dies zu Verklemmungssituationen führen.

Zur Erkennung von Verklemmungen werden im entwickelten Verfahren Situationen durch Merkmale beschrieben. Die Merkmale sind reellwertige Vektoren, die aus den verarbeiteten Sensordaten des Roboters extrahiert werden. Im Grunde beschreiben sie den Aufbau der Umgebung entlang der Bewegung des Roboters hinsichtlich von möglichen Engstellen und Platzkonflikten mit Personen in der Nähe des Roboters. Als Teil der Merkmale wird

die Bewegung von Personen prädiziert, um Platzkonflikte vorherzusagen, die möglicherweise zu Verklemmungen führen können. Jede Situation wird somit zu einem reellwertigen Merkmalsvektor reduziert. Indem die Vektoren zu Klassen zusammengefasst werden, welche die Behandlung ihrer entsprechenden Situation repräsentieren, kann die Situationserkennung als ein Klassifikationsproblem betrachtet werden. Folglich kann ein Klassifikator dazu eingesetzt werden, jeden Merkmalsvektor direkt auf die Behandlung der entsprechenden Situation abzubilden. Eine schematische Darstellung des Verfahrens ist in Abbildung 2.1 zu erkennen. Für die Engstellen der Klinikumgebung könnten die Situationen in drei wesentliche Klassen unterteilt werden.

1. Proceed - Unkritische Situationen

Diese Klasse enthält alle Situationen, die zu keinen Verklemmungen führen und deshalb keine Behandlung benötigen.

2. Parking - Parksituation

Diese Situation entsteht, wenn der Roboter sich durch eine Engstelle bewegen möchte und gleichzeitig eine Person die Engstelle in entgegengesetzter Richtung passiert (siehe Abb. 2.2a). In diesem Fall sollte der Roboter eine geeignete Parkposition aufsuchen und der entgegenkommenden Person die Vorfahrt gewähren.

3. Queueing - Einreihungssituation

Wenn der Roboter und eine Person die gleiche Bewegungsrichtung haben und zur selben Zeit sich durch die Engstelle bewegen möchten, entsteht eine Einreihungssituation (siehe Abb. 2.2b). Zur Behandlung der Situation leitet der Roboter ein Folgen der Person ein, um sich gemeinsam mit der Person durch die Engstelle zu bewegen.

2.2 Formulierung des Lösungsansatzes

Im Folgenden wird der entwickelte Ansatz zur Situationserkennung hinsichtlich des Robotersystems von ROREAS formuliert. Dazu werden zuerst die verwendeten Strukturen des Robotersystems beschrieben, um danach den Ansatz einzuführen.

Damit eine sichere Navigation in der dynamischen Umgebung einer Klinik gewährleistet werden kann, passt das Robotersystem von ROREAS seine Planung ständig den aktuellen Zustand der Umgebung an. Dazu wird der Umgebungszustand periodisch erfasst und für

jeden neu erfassten Zustand ein Plan zu einer Zielposition erzeugt. Es wird angenommen, dass der Roboter und Personen sich in einer zweidimensionalen Welt $\mathcal{W} = \mathbb{R}^2$ bewegen. Zu einen Zeitpunkt t kann der erfasste Umgebungszustand als Tupel

$$E^{(t)} = (M^{(t)}, H^{(t)}) \quad (2.1)$$

mit der Belegheitskarte $M^{(t)} : \mathbb{R}^2 \rightarrow [0..1]$ und der Menge von den erkannten Personen $H^{(t)} = \{h_1^{(t)}, h_2^{(t)}, \dots, h_m^{(t)}\}$ dargestellt werden. Eine Person $h^{(t)} \in H^{(t)}$ wird definiert als

$$h^{(t)} = (\mathbf{p}^{(t)}, \mathbf{v}^{(t)}). \quad (2.2)$$

$\mathbf{p}^{(t)} \in \mathbb{R}^2$ gibt die Position der Person an. $\mathbf{v}^{(t)} \in \mathbb{R}^2$ ist der geschätzte Geschwindigkeitsvektor und gibt die Bewegung der Person pro Sekunde an. Ausgehend von den erfassten Zustand der Umgebung zum Zeitpunkt t plant der Roboter seine Bewegung zu einer Zielposition. Dieser Plan kann als

$$P^{(t)} = (v^{(t)}, \pi^{(t)}) \quad (2.3)$$

definiert werden. $v^{(t)}$ ist der Geschwindigkeitswert des Roboters und gibt die Entfernung an, die der Roboter in einer Sekunde zurücklegt. $\pi^{(t)} = (\mathbf{p}_i \in \mathbb{R}^2)_{i=1, \dots, k}$ repräsentiert den geplanten Bewegungspfad und ist ein Tupel, das aus den zeitlich geordneten Punkten auf dem Bewegungspfad besteht. Für die Punkte \mathbf{p}_j und \mathbf{p}_{j+1} gilt, dass der Roboter \mathbf{p}_j vor \mathbf{p}_{j+1} besucht. Mit (2.1)(2.3) können die betrachteten Situationen dieser Arbeit als

$$S^{(t)} = (E^{(t)}, P^{(t)}) \quad (2.4)$$

definiert werden.

Mit den Informationen der Umgebung und des geplanten Roboterpfades können Engstellsituationen bewertet werden, indem sie nach den identifizierten Klassen aus Abschnitt 2.1 zugeordnet werden. Die Klassen repräsentieren dabei die geeignete Aktion zur Behandlung der Situation. Im verfolgten Ansatz dieser Arbeit werden dazu Situationen durch reellwertige Merkmalsvektoren repräsentiert. Jede Situation wird zuerst durch eine Merkmalsextraktion

$$F : \mathcal{S} \rightarrow \mathbb{R}^n \quad (2.5)$$

auf einen n -dimensionalen Vektor reduziert, wobei \mathcal{S} die Menge aller Situationen bezeichnet. Mit Hilfe eines Klassifikators

$$C : \mathbb{R}^n \rightarrow \{\textit{Proceed}, \textit{Parking}, \textit{Queuing}\} \quad (2.6)$$

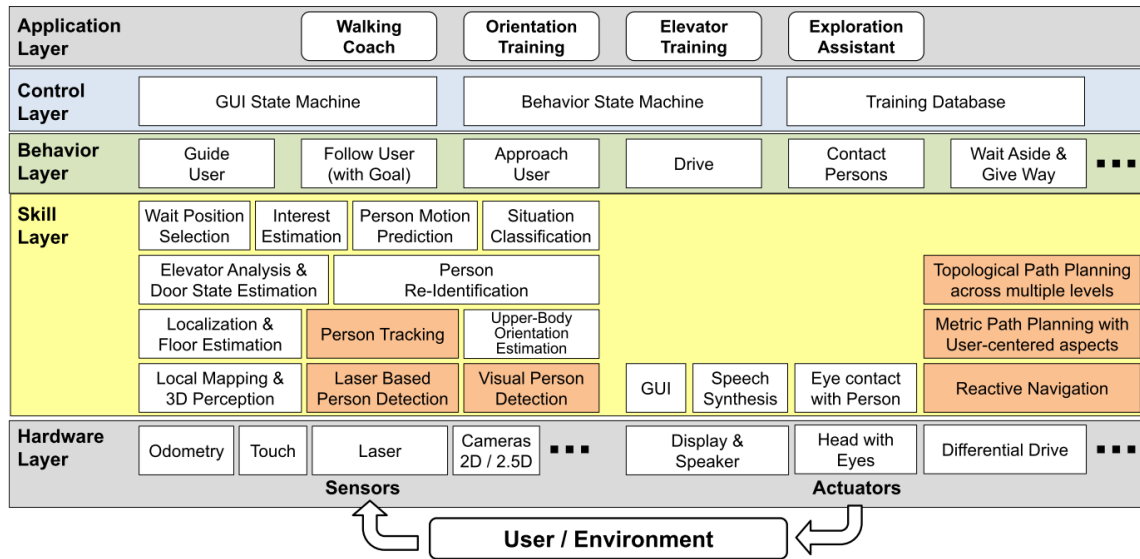


Abbildung 2.3: Darstellung der Schichtenarchitektur des Softwaresystems von ROREAS [GROSS et al., 2014]. Die *Hardware Layer* befindet sich auf der tiefsten Ebene. Mit jeder Schicht darüber wächst der Abstraktionsgrad bezüglich der zu realisierenden Anwendung. Die Schicht mit der höchsten Abstraktion ist die *Application Layer* und kann als Interface zwischen Benutzer und Roboter betrachtet werden. Das entwickelte Verfahren wurde in das Modul *Situation Classifier* integriert.

werden die Merkmalsvektoren den identifizierten Situationsklassen zugeordnet. Folglich kann zu jedem Zeitpunkt t die Situation $S^{(t)} \in \mathcal{S}$ durch Zuordnung zu einer Klasse bewertet werden, um anschließend eine geeignete Behandlung einzuleiten.

2.3 Einordnung in das Softwaresystem von ROREAS

Zur Realisierung des Softwaresystems wurde die Middleware für Robotische Applikationen [EINHORN et al., 2012] verwendet. Das System von ROREAS besteht aus mehreren miteinander kommunizierenden Modulen, die jeweils eine bestimmte Funktion des Roboters kapseln. Diese Module können in eine Schichtenarchitektur angeordnet werden. Die Schichten repräsentieren dabei unterschiedliche Abstraktionsgrade der zu realisierenden Anwendung. In Abbildung 2.3 ist die Schichtenarchitektur von ROREAS dargestellt.

Auf der tiefsten Ebene befindet sich die *Hardware Layer* bestehend aus den Sensoren und Aktoren des Roboters. Darüber befindet sich die *Skill Layer*. Die Module dieser Schicht verarbeiten die Low-Level Informationen der Sensoren und steuern die Aktoren der Hardwa-

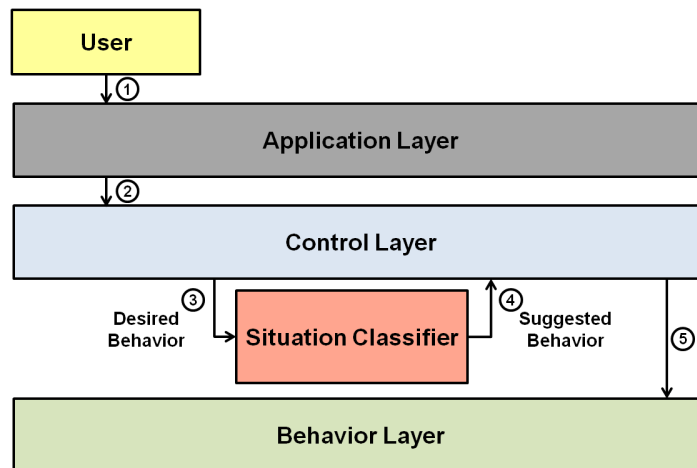


Abbildung 2.4: (1) Nutzer wählt ein Programm der Application Layer aus, welche (2) eine Aktivierung der Zustandsmaschinen in der Control Layer aktiviert. (3) Entsprechend dem ausgewählten Programm teilt die Control Layer dem SC sein gewünschtes Verhalten (Desired Behavior) mit. (4) Der Situation Classifier schlägt davon ausgehend ein Verhalten (Suggested Behavior) vor, welches der Situation der Umwelt angepasst ist. (5) Die Control Layer aktiviert mit diesen Informationen ein Verhalten.

re Layer und realisieren die Hauptfunktionen eines autonom interagierenden Roboters. Zu diesen Funktionen gehören unter anderen die Navigation, das Mapping oder das Personen-tracking. Alleine betrachtet können diese Funktionen kein intelligentes Verhalten realisieren. Dies geschieht erst in der Behavior Layer, die das Zusammenwirken von mehreren Modulen der Skill Layer steuert. Jedes Modul dieser Schicht realisiert ein nach außen sichtbares Verhalten. Dazu zählen z.B. die Zielfahrt, das Folgen oder das Führen einer Person. Die *Control Layer* beinhaltet Zustandsmaschinen zur Ablaufsteuerung der Anwendung, indem bestimmte Verhalten in geordnet aktiviert und deaktiviert werden. Die darauffolgende *Application Layer* enthält die einzelnen Trainingsprogramme des robotischen Lauftrainers und kann als Interface zwischen den Benutzer und Roboter betrachtet werden. [GROSS et al., 2014]

Das entwickelte Verfahren wurde im Modul *Situation Classifier* (SC) der Skill Layer integriert. Der SC kann als ein Berater für die Control Layer betrachtet werden. In Abbildung 2.4 ist dieses Zusammenwirken illustriert. Wenn ein Nutzer ein Trainingsprogramm auswählt, werden entsprechend der Zustandsmaschinen nacheinander Verhalten von der Control Layer aktiviert. Die einzelnen Verhalten werden als *Desired Behavior* bezeichnet. Allerdings sollte je nach den auftretenden Situationen in der Umwelt das Desired Behavior nicht immer ausgeführt werden. Es könnte z.B. bei der Zielfahrt eine Verklemmungssituation mit einer

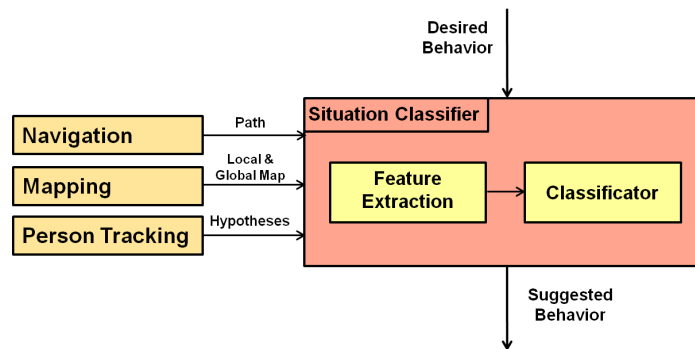


Abbildung 2.5: Blockdiagramm des Situation Classifiers. Zur Extraktion der Merkmale werden die dargestellten Eingaben aus den Modulen der Navigation, Mapping und Personentracking benötigt. Ausgehend von den extrahierten Merkmalen kann mit Hilfe des Klassifikators die Situation bewertet werden, um anschließend das Suggested Behavior auszugeben.

entgegenkommenden Person auftreten. In diesen Fall würde der SC dies vorzeitig erkennen und der Control Layer ein Verhalten zur Behandlung der Situation vorschlagen. Dieses Verhalten ist das sogenannte *Suggested Behavior*.

2.4 Aufbau des Situation Classifiers

Zur Erkennung von Verklemmungssituationen werden in dieser Arbeit Situationen durch reellwertige Merkmalsvektoren dargestellt. Die Merkmale beschreiben im Wesentlichen die Umwelt hinsichtlich von Engstellen und möglichen Platzkonflikten mit den Personen der Umgebung, die entlang der Bewegung des Roboters auftreten können. Um diese Merkmale zu extrahieren werden die verarbeiteten Sensorinformation von den Modulen der Navigation, des Mappings und Personentrackings benötigt. Von den Modulen der Navigation wird der geplante Pfad genutzt, um die Bewegung des Roboters zu bestimmen. Aus den metrischen Karten der Mapping Module kann die Umgebung bezüglich der Position von Hindernissen erfasst werden, die es wiederum erlaubt Engstellen zu erkennen. Damit mögliche Platzkonflikte mit Personen prädiziert werden können, ist es erforderlich deren Position und kinematischen Zustand zu bestimmen. Diese Information wird vom Personentracker geliefert. In Abbildung 2.5 sind die Abhängigkeiten und der schematische Aufbau des SC dargestellt.

2.5 State of the Art

In diesem Abschnitt wird der aktuelle Stand der Forschung im Bereich der expliziten Situationserkennung für die Roboternavigation vorgestellt. Die zusammengetragenen Arbeiten stammen von Recherchen in den Datenbanken *IEEE Xplore* und *Google Scholar* sowie aus einer derzeit laufenden Studentenarbeit am Fachgebiet NIKR. Während der Arbeit konnten nur wenige Verfahren gefunden werden, die eine explizite Situationserkennung für die Roboternavigation einsetzen. Im Bereich der Fahrerassistenzsysteme der Automobilbranche ist die explizite Situationserkennung weitaus verbreiteter. Weil dies den Rahmen dieser Arbeit überschreitet, wird auf das Survey [RENDON-VELEZ et al., 2009] verwiesen.

In [BECK et al., 2011] wird ein allgemeines Framework zur Situationserkennung für mobile Roboter beschrieben. Situationen werden als Muster in den Sensordaten aufgefasst. Auf den Sensordaten wird ein Clustering ausgeführt, welches ähnliche Daten in einem Cluster gruppiert. Die Cluster werden als Zustände in einem *Extensible Markov Model* genutzt, um die raumzeitlichen Beziehungen zu modellieren und später zu erkennen.

Die Arbeit von [TRANBERG HANSEN et al., 2009] beschreibt ein Verfahren zur Erkennung der Interaktionsintention einer Person mit dem Roboter. Dazu werden die Situationen durch Merkmale beschrieben und es wird eine Datenbank aufgebaut, welche die zu erkennen- den Situationen enthält. Die Situationen werden dann durch Matching mit den Fällen der Datenbank erkannt.

In [AYARI et al., 2010] wird ein *Fuzzy-Regelwerk* genutzt, um den Roboter durch eine Umgebung mit anderen beweglichen Entitäten zu navigieren, sodass Platzkonflikte minimiert werden. Die Situationen werden durch die Fuzzy-Regeln erfasst und direkt auf Steuerkommandos abgebildet. Das entwickelte Verfahren wurde in Simulationen evaluiert.

In [DAKULOVIC et al., 2011] wird, wie in der vorliegenden Arbeit, ein Verfahren zur Erkennung von Verklemmungssituationen beschrieben, die durch Engstellen verursacht werden. Engstellen werden analytisch mit Hilfe der Belegheitskarte bestimmt. Sie werden als Minima in der Belegheitskarte beschrieben, die sich entlang einer Achse ausbreiten. Verklemmungssituationen werden erkannt, indem überprüft wird, ob der globale Pfad durch ein Hindernis in einer Engstelle blockiert wird. Wenn eine Verklemmungssituation erkannt wird, sucht der Roboter eine Parkposition auf.

Das entwickelte Verfahren dieser Arbeit nutzt eine explizite Situationserkennung, um Verklemmungen zu erkennen. Sie verfolgt einen ähnlichen Ansatz wie [DAKULOVIC et al., 2011],

nutzt aber die Bewegungsinformation vom Roboter und der Person, um mögliche Konflikte im Voraus zu erkennen. Dies geschieht schon bevor der globale Pfad blockiert wird. Wenn eine Verklemmung erkannt wird, muss zusätzlich zwischen einer Einreihungssituation und einer Parksituation unterschieden werden.

Kapitel 3

Merkmalsextraktion

In diesem Kapitel werden die verwendeten Merkmale zur Beschreibung von Engstellensituationen und die wichtigsten Verarbeitungsschritte zur Extraktion vorgestellt. Die Merkmale können in zwei Gruppen eingeteilt werden: Beschreibung von prädizierten Platzkonflikten mit Personen (Abschnitt 3.1) und Beschreibung von detektierten Engstellen in Verbindung mit den prädizierten Platzkonflikten (Abschnitt 3.2).

3.1 Lineare Prädiktion von Platzkonflikten

3.1.1 Merkmale

Innerhalb der Engstellen auf den Klinikkorridoren ist es nicht möglich, dass gleichzeitig zwei Personen nebeneinander Platz haben. Wenn durch die Bewegung des Roboters und einer Person Platzkonflikte innerhalb der Engstelle entstehen, führt dies zu Verklemmungen. Durch die Prädiktion dieser Platzkonflikte können Verklemmungssituationen im Voraus erkannt und behandelt werden. Zur prädiktiven Erkennung von Platzkonflikten werden die Kollisionen zwischen den raumzeitlichen Bewegungstrajektorien des Roboters und einer Person bestimmt. Für die weiteren Betrachtungen wird angenommen, dass der Roboter und die Person durch Kreisformen mit konstanten Radien repräsentiert werden und ein lineares Bewegungsmodell haben. Kollisionen können bestimmt werden, indem der Roboter und die Person gleichförmig entlang des geplanten Bewegungspfades bzw. des geschätzten Geschwindigkeitsvektors bewegt werden. Eine Kollision tritt genau dann auf, wenn ein Zeitpunkt existiert, in dem es zu einer Überschneidung der Kreisformen des Roboters und der Person kommt. In Abbildung 3.1 ist die Prädiktion schematisch dargestellt.

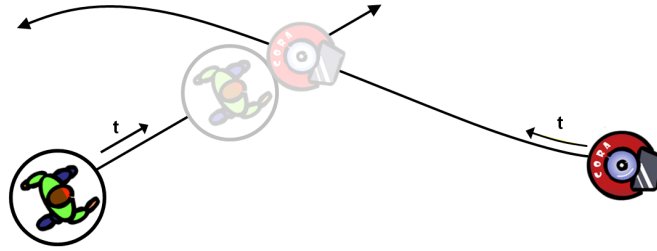


Abbildung 3.1: Schema zur Prädiktion von Platzkonflikten. Die Platzkonflikte werden durch Kollisionen auf den raumzeitlichen Bewegungstrajektorien des Roboters und der Person beschrieben. Für den Roboter und die Person wird eine Kreisform angenommen. Zur Bestimmung der Kollisionen wird der Roboter und die Person gleichförmig entlang des Roboterpfades bzw. Geschwindigkeitsvektors bewegt.

Die Kollisionen mit dem Roboter werden getrennt für jede Person bestimmt. Ausgehend davon wird für jede Person ein Merkmalsvektor extrahiert. Die Merkmale beziehen sich dabei auf die *zeitlich nächste Kollision* der Person. Im Wesentlichen beschreiben sie die raumzeitliche Lage der Person und des Roboters zum Zeitpunkt der Kollision. In Abbildung 3.2 werden die verwendeten Merkmale zur Beschreibung von Platzkonflikten erläutert.

3.1.2 Bestimmung der Trajektorien

Die Trajektorie einer Entität kann als eine Funktion

$$\tau : [0, T_{max}] \rightarrow \mathbb{R}^2 \quad (3.1)$$

über die Zeit betrachtet werden. Die Zeit sei in Einheiten von Sekunden angegeben. T_{max} gibt den maximal betrachteten Zeithorizont an. Unter der erwähnten Annahme einer Kreisform für die Entitäten beschreibt τ die Trajektorie des Kreismittelpunktes. Zur Bestimmung von Kollisionen zwischen Trajektorien wird in dieser Arbeit eine parametrische Darstellung verwendet, die aus den Personenzustand und der geplanten Roboterbewegung unter Annahme eines linearen Bewegungsmodell berechnet werden. Die berechnete Trajektorie gibt die Bewegung relativ zum Zeitpunkt des erfassten Personenzustandes bzw. der geplanten Roboterbewegung an.

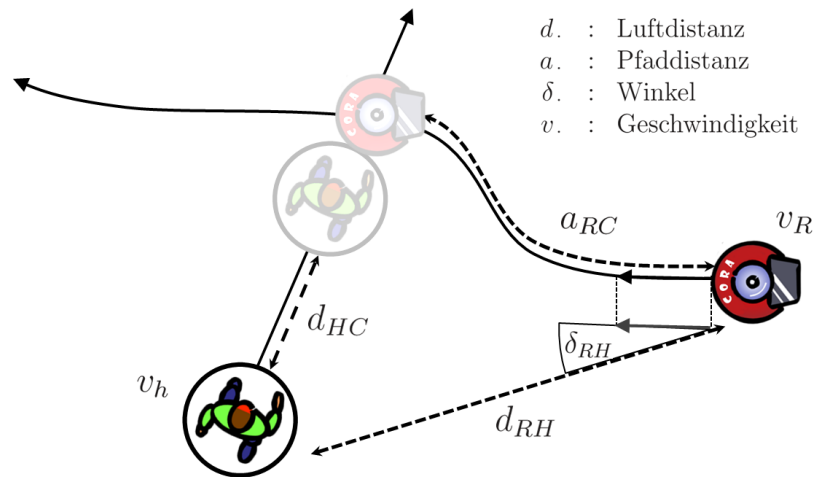


Abbildung 3.2: Für jede Person wird ein Merkmalsvektor zur Beschreibung der Platzkonflikte mit dem Roboter extrahiert. Platzkonflikte werden durch Kollisionen der Person mit dem Roboter beschrieben. Dabei wird von jeder Person nur die zeitlich nächste Kollision betrachtet. Die Merkmale zur Beschreibung der Platzkonflikte sind:

- a_{RC} : Pfaddistanz des Roboters zur Kollision.
- d_{HC} : Luftdistanz der Person zur Kollision.
- d_{RH} : Luftdistanz des Roboters zur kollidierenden Person.
- δ_{RH} : Winkel des Aufenthaltspunktes der Person relativ zur aktuellen Bewegungsrichtung des Roboters.
- v_R : Geschwindigkeit des Roboters.
- v_h : Geschwindigkeit der Person.

Es ist zu beachten, dass Pfeile entlang des Pfades, die Pfaddistanz und nicht die Luftdistanz darstellen.

Trajektorien von Personen

Der erfasste Zustand einer Person ist als Tupel $h = (\mathbf{p}, \mathbf{v})$ (siehe Gleichung 2.2) definiert, wobei \mathbf{p} die momentane Position und \mathbf{v} der geschätzte Geschwindigkeitsvektor ist. Für die Bestimmung der Trajektorie einer Person wird ein lineares Bewegungsmodell angenommen. Als Folge wird die Person gleichförmig entlang ihres Geschwindigkeitsvektor \mathbf{v} bewegt. Die Trajektorie einer Person kann durch

$$\begin{aligned} \tau_h &: [0, \infty) \rightarrow \mathbb{R}^2 \\ \tau_h(t) &= t \mathbf{v} + \mathbf{p} \end{aligned} \tag{3.2}$$

bestimmt werden.

Trajektorie des Roboters

Die geplante Roboterbewegung ist durch $P = (v, \pi)$ (siehe Gleichung 2.3) gegeben. v gibt die Geschwindigkeit an und $\pi = (\mathbf{p}_i)_{i=1, \dots, k}$ ist das Tupel, welches die zeitlich geordneten Punkte des Bewegungspfades enthält. Mit den Pfadpunkten aus π kann zur Bestimmung der Trajektorie ein Interpolationsansatz mit einem *Spline* verwendet werden. Ein Spline ist eine Funktion, die stückweise aus Polynomen besteht. Unter Annahme eines linearen Bewegungsmodell werden Polynome ersten Grades verwendet. Die Funktion zwischen zwei aufeinanderfolgenden Punkten \mathbf{p}_j und \mathbf{p}_{j+1} wird dann zu einer Geraden.

Zur Interpolation einer raumzeitlichen Trajektorie muss jedem Punkt \mathbf{p}_i eine Zeit t_i zugeordnet werden, die den Aufenthaltszeitpunkt des Roboters für diesen Punkt angibt. Diese Zeitpunkte können mit folgender Vorschrift berechnet werden.

$$\begin{aligned} \mathbf{p}_1 &: t_1 = 0 \\ \mathbf{p}_i &: t_i = t_{i-1} + \frac{\|\mathbf{p}_i - \mathbf{p}_{i-1}\|}{v}, \quad 2 \leq i \leq k \end{aligned} \tag{3.3}$$

$\|\cdot\|$ bezeichne dabei die euklidische Norm. Die Aufenthaltszeitpunkte werden zum Tupel $\chi = (t_i)_{i=1, \dots, k}$ zusammengefasst. Mit den Aufenthaltszeitpunkten kann die lineare Trajektorie des Roboters bestimmt werden durch:

$$\begin{aligned} \tau_R &: [0, t_k) \rightarrow \mathbb{R}^2 \\ \tau_R(t) &= \frac{t - t_i}{t_{i+1} - t_i} (\mathbf{p}_{i+1} - \mathbf{p}_i) + \mathbf{p}_i, \quad t \in [t_i, t_{i+1}) \text{ mit } 1 \leq i \leq k \end{aligned} \tag{3.4}$$

3.1.3 Bestimmung der Kollisionen

Trajektorie der Kreise

Der Roboter und die Personen werden durch Kreisformen mit konstanten Radien repräsentiert. Die Trajektorie der Kreisformen kann als Funktion

$$\begin{aligned} \mathcal{A}(t) &\subseteq \mathbb{R}^2, & t &\in [0, T_{max}] \\ \mathcal{A}(t) &= \{ \mathbf{p} \in \mathbb{R}^2 \mid \|\tau(t) - \mathbf{p}\| \leq \rho \} \end{aligned} \quad (3.5)$$

der eingenommenen Kreisregion betrachtet werden, wobei ρ der Radius und $\tau(t)$ die Trajektorie des Kreismittelpunktes der betrachteten Entität ist. $\mathcal{A}(t)$ ist also die Menge der Punkte zum Zeitpunkt t , die sich innerhalb der Kreisfläche befindet. Im Folgenden bezeichne \mathcal{A}_R und \mathcal{A}_h die Trajektorie des Kreises des Roboters bzw. der Person. Sei dazu ρ_R der Roboterradius und ρ_h der Radius einer Person.

Kollisionsbedingung

Eine Kollision des Roboters mit einer Person kann als Zeitpunkt t_C definiert werden, in denen sich die eingenommenen Kreisregionen des Roboters mit der Person überschneiden. In diesem Zeitpunkt gilt

$$\mathcal{A}_R(t_C) \cap \mathcal{A}_h(t_C) \neq \emptyset. \quad (3.6)$$

Für eine Vereinfachung der weiteren Betrachtungen wird Gleichung (3.6) mit einem weit verbreiteten Ansatz umgeformt, sodass die direkte Bestimmung des Durchschnitts nicht benötigt wird. In diesem Ansatz wird die Kreisform des Roboters auf einen Punkt reduziert und der Personenradius um den Roboterradius erweitert. Sei $\rho'_h = \rho_R + \rho_h$ der modifizierte Personenradius und \mathcal{A}'_h die modifizierte Trajektorie der Person. Da der Roboter auf einen Punkt reduziert wurde, vereinfacht sich die Trajektorie des Kreises des Roboters zur Trajektorie des Kreismittelpunktes. Für Gleichung (3.6) gilt dann:

$$\begin{aligned} \mathcal{A}_R(t_C) \cap \mathcal{A}_h(t_C) &\neq \emptyset \\ \Leftrightarrow \tau_R(t_C) &\in \mathcal{A}'_h(t_C) \end{aligned} \quad (3.7)$$

Sie reduziert sich zu einer einfachen Bestimmung eines Punktes im Kreis.

Sampling-basierter Ansatz

Beim Sampling-basierten Ansatz werden die Punkte des geplanten Bewegungspfad $\pi = (\mathbf{p}_i)_{i=1,\dots,k}$ mit den Aufenthaltszeitpunkten $\chi = (t_i)_{i=1,\dots,k}$ als Samples der Trajektorie

des Roboters betrachtet. Die Kollisionen werden bestimmt, indem für jeden dieser Samples die Kollisionsbedingung mit der Trajektorie \mathcal{A}'_h der Person (Gleichung 3.6) überprüft wird. Algorithmus 1 beschreibt diesen Ansatz.

Input: Bewegungspfad $\pi = (p_i)_{i=1,\dots,k}$, Aufenthaltszeitpunkte $\chi = (t_i)_{i=1,\dots,k}$,
 Modifizierte Trajektorie des Kreises \mathcal{A}'_h

Output: Kollisionszeitpunkte C_h

```

1  $C_h \leftarrow \emptyset$ ;
2 for  $i \leftarrow 1$  to  $k$  do
3   if  $\mathbf{p}_i \in \mathcal{A}'_h(t_i)$  then
4      $C_h \leftarrow C_h \cup \{t_i\}$ ;
5   end
6 end

```

Algorithmus 1: Sampling-basierter Ansatz zur Bestimmung von Kollisionen

Der Vorteil des Sampling-basierten Ansatz ist, dass keine Annahmen über die Form des Roboters oder der Person angenommen werden braucht. Der Test in Zeile 3 könnte durch eine beliebige Kollisionsbedingung ersetzt werden. Zudem muss keine parametrische Form der Personentrajektorie vorhanden sein. Diese könnte auch durch Samples repräsentiert werden. Bei den verwendeten Samples dieser Arbeit ist allerdings die Güte des Ergebnisses von den Abstand der Punkte aus π abhängig. Wenn die Punkte zu weit voneinander entfernt sind, können dadurch Kollisionen unentdeckt bleiben.

Analytischer Ansatz

Unter der Annahme einer linearen Bewegung kann der geplante Pfad des Roboters als linearer Spline dargestellt werden. Die Bestimmung dieser Trajektorie ist in Abschnitt 3.1.2 (Trajektorie des Roboters) beschrieben. Durch die Benutzung der parametrischen Darstellung der Trajektorien vom Roboter und der Person können Kollisionen analytisch bestimmt werden. Damit ist es auch möglich eine Subsamplegenauigkeit zu erreichen.

Beim analytischen Ansatz kann die modifizierte Kollisionsbedingung (Gleichung 3.7) für jedes Funktionsstück im Zeitintervall $[t_i, t_{i+1})$ der Robotertrajektorie τ_R in eine quadratische

<p>Input: Robotertrajektorie τ_R, Modifizierte Trajektorie der Person \mathcal{A}'_h</p> <p>Output: Kollisionsintervalle C_h</p> <pre> 1 $C_h \leftarrow \emptyset$; 2 for $i \leftarrow 1$ to $k - 1$ do 3 Bestimme Koeffizienten a_i, b_i, c_i nach Gl. (3.8); 4 Berechne Zeitintervalle $L = \{l_1, \dots, l_n\}$ als Lösung der Ungleichung $a_i t^2 + b_i t + c_i \leq 0$; 5 for $j \leftarrow 1$ to n do 6 $l \leftarrow l_j \in L$; 7 $l' \leftarrow l \cap [t_i, t_{i+1})$; 8 if $l' \neq \emptyset$ then 9 $C_h \leftarrow C_h \cup \{l'\}$ 10 end 11 end 12 end </pre>
--

Algorithmus 2: Analytischer Ansatz zur Bestimmung von Kollisionen

Ungleichung überführt werden:

$$\begin{aligned}
& \tau_R(t_C) \in \mathcal{A}'_h(t_C) \\
& \stackrel{(Gl. 3.5)}{\Leftrightarrow} \|\tau_R(t_C) - \tau_h(t_C)\| \leq \rho'_h \\
& \stackrel{t_C \in [t_i, t_{i+1})}{\Leftrightarrow} \left\| \frac{t_C - t_i}{t_{i+1} - t_i} (\mathbf{p}_{i+1} - \mathbf{p}_i) + \mathbf{p}_i - \tau_h(t_C) \right\| \leq \rho'_h \\
& \Leftrightarrow a_i t_C^2 + b_i t_C + c_i \leq 0
\end{aligned} \tag{3.8}$$

Anders betrachtet, können mit Gleichung (3.8) für jedes Funktionsstück die Zeitintervalle berechnet werden, in denen eine Kollision mit der Person vorliegt. Algorithmus 2 beschreibt dieses Vorgehen. Es ist zu beachten, dass eine quadratischen Ungleichung mehrere Zeitintervalle als Lösung haben kann.

Mit dem analytischen Ansatz können im Gegensatz zum sampling-basierten Ansatz auch Kollisionen zwischen zwei Pfadpunkten bestimmt werden. Allerdings ist die Annahme eines linearen Bewegungsmodells zwischen zwei Pfadpunkten eine grobe Abschätzung der Roboterbewegung. Mit größer werdenden Abständen zwischen aufeinander folgenden Pfadpunkten steigt die Wahrscheinlichkeit, dass die tatsächliche Roboterbewegung vom linearen Modell abweicht. Somit wächst auch die Wahrscheinlichkeit einer Fehldetektion.

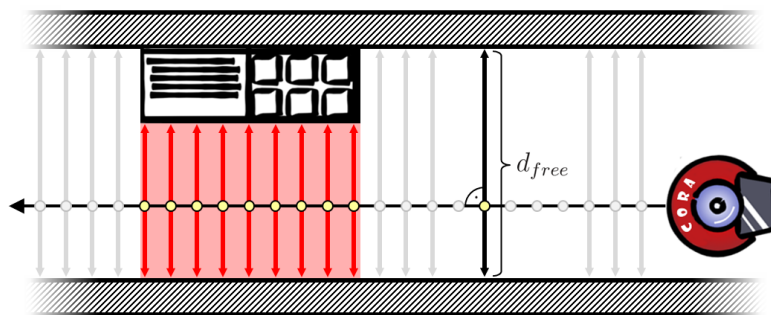


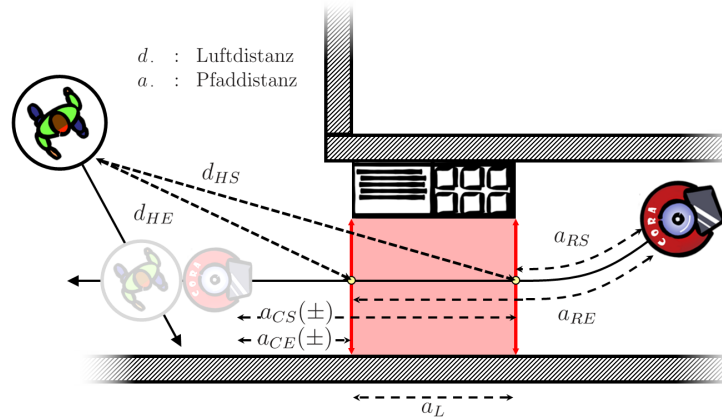
Abbildung 3.3: Darstellung des Verfahren zur Engstellenerkennung. Auf dem geplanten Roboterpfad werden für diskrete Punkte Normalen berechnet. Diese verlaufen senkrecht zum geplanten Pfad. Entlang dieser Normalen wird mit der Belegheitskarte der Umgebung die Durchfahrtsbreite bestimmt. Engstellen sind dann Pfadabschnitte, in denen die maximale Durchfahrtsbreite einen Schwellwert unterschreitet. d_{free} bezeichnet die Durchfahrtsbreite der dargestellten Normale. Der rot markierte Pfadabschnitt stellt eine Engstelle dar.

3.2 Engstellenerkennung

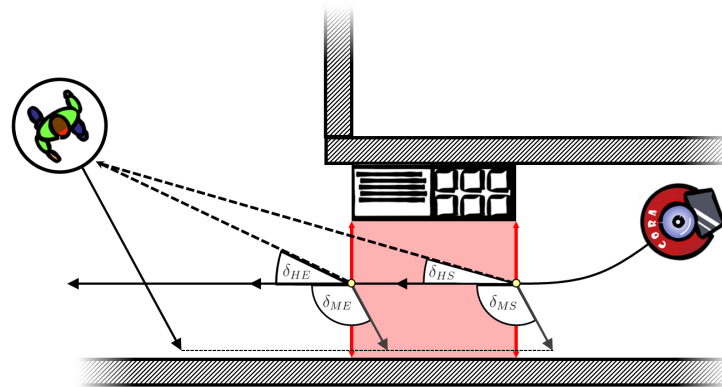
3.2.1 Merkmale

Für die Erkennung von Engstellen wurde ein Verfahren aus vorangegangenen Arbeiten des Fachgebietes NIKR verwendet und weiter verfeinert. Das Verfahren berechnet für diskrete Punkte entlang des geplanten Bewegungspfades des Roboters Normalen, die senkrecht zum Pfad verlaufen. Entlang der Normalen wird mit Hilfe der Belegheitskarte der Umgebung die Distanz zu den Hindernissen berechnet. Diese Distanzen können als Durchfahrtsbreite für den Roboter betrachtet werden. Engstellen können als zusammenhängende Pfadabschnitte beschrieben werden, in denen die durchgängige Durchfahrtsbreite nur Platz für den Roboter bzw. eine Person zulässt. In Abbildung 3.3 ist das Verfahren schematisch dargestellt.

Die Merkmale zur Beschreibung der Engstelle beziehen sich auf die *zeitlich nächste Engstelle* auf dem Bewegungspfad. Die Merkmale werden in Zusammenhang mit den prädizierten Platzkonflikten der Personen extrahiert. Dabei wird von jeder Person nur die zeitlich nächste Kollision betrachtet. Folglich wird für jede Person ein Merkmalsvektor extrahiert. Die Merkmale beschreiben im Grunde die Lage der Engstelle zum Roboter und jeder Person mit Platzkonflikten. In der Abbildungen 3.4 werden die Merkmale erläutert.



(a) Distanzen zur Engstelle



(b) Orientierung zur Engstelle

Abbildung 3.4: Engstellen sind Abschnitte auf dem geplanten Roboterpfad, in denen die Durchfahrtsbreite begrenzt ist, sodass nur Platz für eine Person ist. Die entwickelten Merkmale betrachten nur die zeitlich nächste Engstelle in Zusammenhang mit den zeitlich nächsten Kollisionen von den Personen mit dem Roboter. Für jede Person wird folglich ein Merkmalsvektor extrahiert. In a und b sind die verwendeten Merkmale dargestellt:

- d_{HS}, d_{HE} : Luftdistanz der Person zum Engstellenstart bzw. -ende.
- a_{RS}, a_{RE} : Pfaddistanz des Roboters zum Engstellenstart bzw. -ende.
- a_{CS}, a_{CE} : Pfaddistanz der Kollision relativ zum Engstellenstart bzw. -ende. Sei a_{RC} die Pfaddistanz des Roboters zu Kollision. Dann gilt $a_{CS} = a_{RC} - a_{RS}$ und $a_{CE} = a_{RC} - a_{RE}$.
- a_L : Länge der Engstelle.
- δ_{HS}, δ_{HE} : Winkel der Orientierung der Person relativ zur Bewegungsrichtung des Roboters am Engstellenstart bzw. -ende.
- δ_{MS}, δ_{ME} : Winkel der Bewegungsrichtung der Person relativ zur Bewegungsrichtung des Roboters am Engstellenstart bzw. -ende.

3.2.2 Berechnung der Durchfahrtsbreite

Die Robotertrajektorie τ_R kann neben der Parametrisierung über die Zeit aus (Abschnitt 3.1) auch über die *Bogenlänge* parametrisiert werden. Die Bogenlänge gibt dabei den zurückgelegten Weg auf der Trajektorie an. Für die folgenden Betrachtungen bezeichne

$$\alpha_R : [0, L] \rightarrow \mathbb{R}^2 \quad (3.9)$$

die über die Bogenlänge parametrisierte Robotertrajektorie.

Ausgehend von der Robotertrajektorie α_R können Normalen in einem Punkt als Geraden angesehen werden, die senkrecht zur Bewegungsrichtung in diesen Punkt verlaufen. Die Robotertrajektorie kann dazu als komponentenweise Funktion

$$\alpha_R(l) = \begin{pmatrix} x(l) \\ y(l) \end{pmatrix}, \quad l \in [0, L] \quad (3.10)$$

betrachtet werden. Die Bewegungsrichtung zu einer Bogenlänge l entspricht der Tangente in diesem Punkt. Wenn α_R im Intervall $[0, L]$ differenzierbar ist, kann die Tangente mit

$$\dot{\alpha}_R(l) = \begin{pmatrix} \dot{x}(l) \\ \dot{y}(l) \end{pmatrix}, \quad \dot{x} = \frac{dx}{dl}, \dot{y} = \frac{dy}{dl} \quad (3.11)$$

berechnet werden [KÜHNEL, 2008]. Daraus kann die Normale mit

$$n_R(l) = \begin{pmatrix} -\dot{y}(l) \\ \dot{x}(l) \end{pmatrix} \quad (3.12)$$

bestimmt werden. Diese Normale ist linksorientiert, d.h. sie breitet sich auf der linken Seite der Trajektorie aus.

Für diskrete Punkte $\alpha_R(l)$ auf der Robotertrajektorie kann die Durchfahrtsbreite bestimmt werden, indem ausgehend von $\alpha_R(l)$ entlang von $n_R(l)$ und $-n_R(l)$ in der Belegheitskarte M ein *Raycasting* ausgeführt wird. Damit kann die Kollision des Strahls mit Hindernissen bestimmt werden und schließlich auch die Distanz zu diesen Hindernissen. Die Summe aus der so ermittelten links- und rechtsseitigen Distanz ergibt die Durchfahrtsbreite. Algorithmus 3 beschreibt dieses Vorgehen.

Input: Belegheitskarte M , Differenzierbare Robotertrajektorie α_R , Schrittweite s

Output: Durchfahrtsbreiten $D = \{(l_1, d_1), \dots, (l_m, d_m)\}$

```

1  $D \leftarrow \emptyset$ ;
2  $l \leftarrow 0$ ;
3 while  $l \leq L$  do
4    $n \leftarrow n_R(l)$ ; // Bestimmung der Normale nach Gl.(3.12)
5   Bestimme Distanzen  $d_{left}$  und  $d_{right}$  zu Hindernissen mit Raycasting in  $M$  entlang
   von  $n$  bzw.  $-n$ ;
6    $D \leftarrow \{(l, d_{left} + d_{right})\}$ ;
7    $l \leftarrow l + s$ ;
8 end

```

Algorithmus 3: Berechnung der Durchfahrtsbreiten entlang der Robotertrajektorie

3.2.3 Trajektorie mit Bogenlängenparametrisierung

Lineare Spline-Interpolation

Für die Berechnung der Durchfahrtsbreite mit dem Algorithmus 3 wird eine über die Bogenlänge parametrisierte und differenzierbare Trajektorie benötigt. Diese kann aus der geplanten Roboterbewegung $P = (v, \pi)$ berechnet werden. Zur Bestimmung dieser Trajektorie kann der Interpolationsansatz aus Abschnitt 3.1.2 (Trajektorie des Roboters) angepasst werden. Anstelle von der Zuordnung der Aufenthaltszeit wird mit

$$\begin{aligned}
 \mathbf{p}_1 & : l_1 = 0 \\
 \mathbf{p}_i & : l_i = l_{i-1} + \|\mathbf{p}_i - \mathbf{p}_{i-1}\|, \quad 2 \leq i \leq k
 \end{aligned} \tag{3.13}$$

jedem Punkt eine Bogenlänge zugeordnet. Für die Robotertrajektorie α_R folgt:

$$\begin{aligned}
 \alpha_R & : [0, l_k) \rightarrow \mathbb{R}^2 \\
 \alpha_R(l) & = \frac{l-l_i}{l_{i+1}-l_i} (\mathbf{p}_{i+1} - \mathbf{p}_i) + \mathbf{p}_i, \quad t \in [l_i, l_{i+1}) \text{ mit } 1 \leq i \leq k
 \end{aligned} \tag{3.14}$$

Der Vorteil von diesem Ansatz ist, dass die Laufzeit linear mit der Anzahl der Pfadpunkte wächst. Da die Trajektorie zwischen den Punkten durch eine Gerade interpoliert wird, ist die erste Ableitung $\alpha'_R(t)$ zwischen zwei Punkten konstant. Es kann somit nicht die Krümmung der Trajektorie berücksichtigt werden. Dies hat zur Folge, dass bei einer kurvigen Bahn des Pfades die resultierenden Normalen große Lücken aufweisen (siehe Abbildung 3.5). Dadurch können Engstellen übersehen werden.

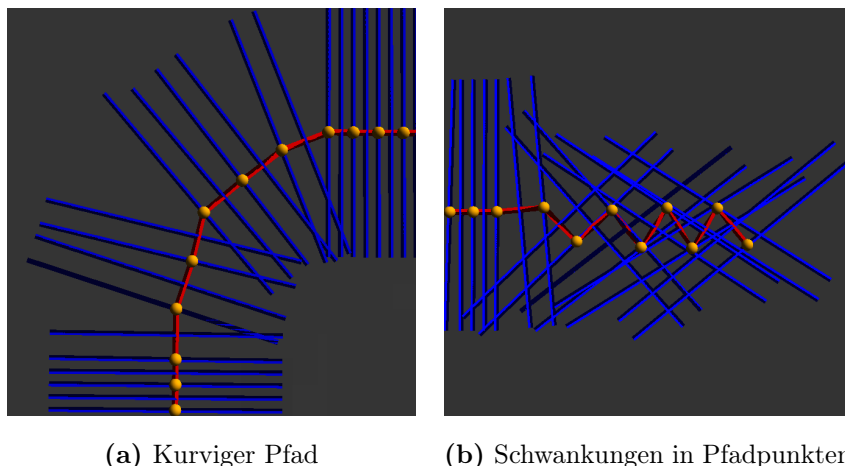


Abbildung 3.5: Die Abbildungen zeigen reale Pfade des Robotersystems. Die gelblichen Kugeln sind die Pfadpunkte. Der davon resultierende lineare Spline ist durch die rote Linie dargestellt. Die blauen Geraden stellen die berechneten Normalen dar. a Auf kurvigen Pfaden können Lücken in der Berechnung der Durchfahrtsbreite entstehen. Dies liegt an der konstanten ersten Ableitung eines linearen Spline, die eine Krümmung des Roboterpades nicht berücksichtigen kann. b Schwankungen in den Pfadpunkten führen wegen der Interpolationsbedingung zu fehlerhaften Normalen, welche die Durchfahrtsbreite verfälscht.

Des Weiteren wurde in Tests festgestellt, dass Schwankungen bei den Pfadpunkten auftreten können, sodass diese nicht immer auf den tatsächlichen Pfad liegen (siehe Abbildung 3.5). Durch den Interpolationsansatz muss zwangsläufig jeder Pfadpunkt auf den resultierenden Spline liegen und es entsteht ein Zick-Zack-Muster in der Trajektorie. Die damit bestimmten Normalen verfälschen die Durchfahrtsbreite.

Kubischer Smoothing Spline

Mit dem kubischen Smoothing Spline können *Curve Fittings* Probleme gelöst werden. Dabei ist eine Menge von Datenpunkten $T = \{(x_i, y_i) \in \mathbb{R}^2\}_{1 \leq i \leq N}$ mit $x_1 < x_2 < \dots < x_n$ gegeben. Es wird versucht ein Spline $S(x)$ bestehend aus stückweisen kubischen Polynomen an T anzupassen. Die Koeffizienten der Polynome werden dazu angepasst, sodass das Gütekriterium

$$p \sum_{i=1}^n (S(x_i) - y_i)^2 + (1 - p) \int_{x_1}^{x_n} \ddot{S}(x)^2 dx, \quad p \in (0, 1) \quad (3.15)$$

minimiert wird.

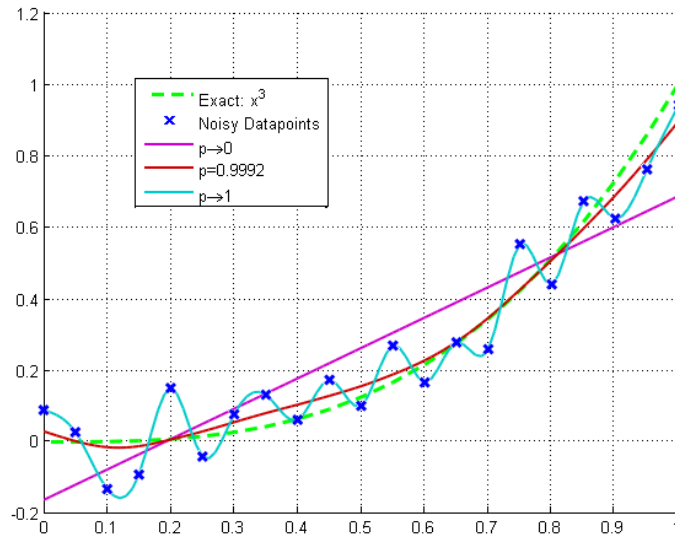


Abbildung 3.6: Darstellung von Kubischen Smoothing Splines für unterschiedliche p -Werte [MATLAB, 2014]. Die Datenpunkte stammen von der Funktion $f(x) = x^3$ und wurden unter Annahme einer Gleichverteilung mit einem zufälligen Wert aus $[-0.15, 0.15]$ verrauscht.

Der erste Summand bestimmt wie stark der resultierende Spline von den gegebenen Datenpunkten abweichen darf. Der Term des Integrals der zweiten Ableitung bestraft dagegen die Nichtlinearität des Splines. Mit dem Parameter p kann der Grad der Glättung kontrolliert werden, indem der Einfluss der beiden Terme abgestimmt wird. In Abbildung 3.6 ist der Effekt von unterschiedlichen Werten für p an einem Beispiel dargestellt. Für $p \rightarrow 1$ konvergiert $S(x)$ zum Spline des Interpolationsproblems. Dagegen konvergiert $S(x)$ für $p \rightarrow 0$ zur Lösung der Methode der kleinsten Quadrate mit einer linearen Funktion. [DE BOOR, 2001]

Zur Implementierung wurde die Bibliothek ALGLIB [ANATOLYEVICH, 2014] verwendet. Mit den kubischen Smoothing Spline können die Probleme der linearen Spline-Interpolation vermieden werden. Schwankungen in den Pfadpunkten sind unproblematisch, weil sie durch geeignete Wahl von p geglättet werden können. Zudem besitzen kubische Spline eine kontinuierliche erste Ableitung. Dadurch passen die Normalen sich der Krümmung der geplanten Bewegungspfade an und es werden auf kurvigen Bahnen die Entstehung von Lücken minimiert. In Abbildung sind

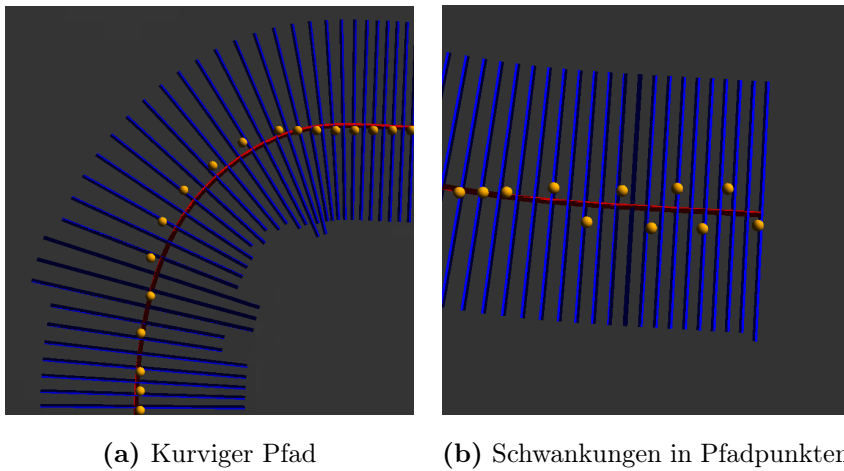


Abbildung 3.7: Die Abbildungen zeigen reale Pfade des Robotersystems. Die gelblichen Kugeln sind die Pfadpunkte. Der davon resultierende kubische Smoothing Spline ist durch die rote Linie dargestellt. Die blauen Geraden stellen die berechneten Normalen dar. Die Probleme der linearen Spline-Interpolation (s. Abb. 3.5) werden vermieden, da die erste Ableitung eines kubischen Spline kontinuierlich ist und keine strikte Interpolationsbedingung eingehalten werden muss.

Kapitel 4

Klassifikation

In diesem Kapitel werden die verwendeten Klassifikatoren dieser Arbeit vorgestellt, die ausgehend von den Merkmalen aus Kapitel 3 die Situationen erkennen. Als Klassifikator wird die *Support Vektor Maschine* (Abschnitt 4.2) und ein manuell erstellter *Entscheidungsbaum* (Abschnitt 4.3) genutzt.

4.1 Einleitung

Jede Situation kann durch einen Merkmalsvektor dargestellt und einer Klasse zugeordnet werden, welche die Aktion zur Behandlung der Situation repräsentiert. Ein Klassifikator kann folglich genutzt werden, um jede Situation direkt auf die geeignete Reaktion abzubilden. Für die Anwendung in der Klinik können drei Klassen von Reaktionen unterschieden werden. Diese wurden im Abschnitt 2.1 beschrieben und werden zur besseren Übersicht im Folgenden wiederholt:

1. **Proceed - Unkritische Situationen**

Diese Klasse enthält alle Situationen, die zu keinen Verklemmungen führen und deshalb keine Behandlung benötigt.

2. **Parking - Parksituation**

Diese Situation entsteht, wenn der Roboter sich durch eine Engstelle bewegen möchte und gleichzeitig eine Person die Engstelle in entgegengesetzter Richtung passiert (siehe Abb. 2.2a). In diesem Fall sollte der Roboter eine geeignete Parkposition aufsuchen und der entgegenkommenden Person die Vorfahrt gewähren.

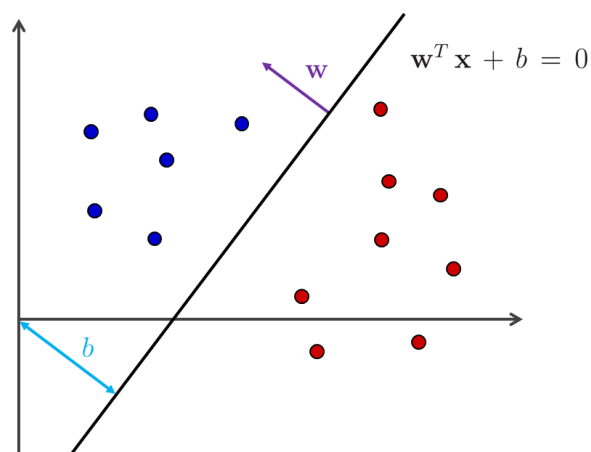


Abbildung 4.1: Die SVM lernt ausgehend von einem gelabelten Datensatz eine Hyperebene zur Klassentrennung. Im dargestellten zweidimensionalen Fall wird eine Trenngerade gelernt.

3. Queueing - Einreihungssituation

Wenn der Roboter und eine Person die gleiche Bewegungsrichtung haben und zur selben Zeit sich durch die Engstelle bewegen möchten, entsteht eine Einreihungssituation (siehe Abb. 2.2b). Zur Behandlung der Situation leitet der Roboter ein Folgen der Person ein, um sich gemeinsam mit der Person durch die Engstelle zu bewegen.

Eine Übersicht der verwendeten Merkmale ist in den Abbildungen 3.2 und 3.4 zu finden. Die Merkmale werden getrennt für jede erkannte Person in der Situation berechnet. Für jede Person wird also ein Merkmalsvektor extrahiert und durch den Klassifikator einer Reaktion zugeordnet. Um die endgültige Handlung des Roboters zu bestimmen, wird die Klasse der Person mit der zeitlich nächsten Kollision ausgewählt

4.2 Support Vektor Maschine

Die Support Vektor Maschine (SVM) ist ein maschinelles Lernverfahren zum Lösen des binären Klassifikationsproblems. Es lernt ausgehend von einem gelabelten Datensatz

$$(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_N, y_N), \quad \mathbf{x}_i \in \mathbb{R}^n, \mathbf{y}_i \in \{-1, 1\} \text{ mit } 1 \leq i \leq N \quad (4.1)$$

eine Hyperebene der Form

$$f(\mathbf{x}) = \mathbf{w}^T \mathbf{x} + b, \quad \mathbf{w}, \mathbf{x} \in \mathbb{R}^n, b \in \mathbb{R}. \quad (4.2)$$

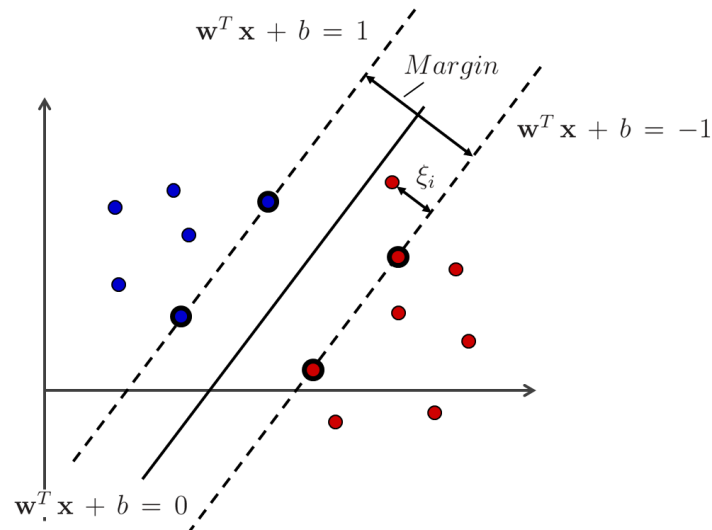


Abbildung 4.2: Beispiel eines Soft-Margin und die Support Vektoren einer Gerade. Der Margin ist der Abstand zwischen den Grenzen einer Gerade. Die Grenzen sind im Beispiel durch gestrichelte Linien dargestellt. Die Support Vektoren sind die hervorgehobenen Datenpunkte auf den Grenzen der Gerade.

Diese Hyperebene wird zur Trennung der Klassen genutzt. Unbekannte Datenpunkte können durch

$$y = \text{sgn}(f(\mathbf{x})) \quad (4.3)$$

klassifiziert werden. In Abbildung 4.1 ist das Grundprinzip der Klassentrennung dargestellt.

Für die Trennung von Klassen existieren eine unendliche Anzahl von Hyperebenen. Damit die beste Hyperebene ausgewählt werden kann, wird ein Gütekriterium benötigt. Die SVM definiert dafür den sogenannten *Margin*, der die Distanz zu den Datenpunkten an den Grenzen angibt. Die Hyperebene mit dem größten Margin wird als Trennebene verwendet. Aus der Definition des Margins ergibt sich das quadratische Optimierungsproblem

$$\min_{\mathbf{w}} \frac{1}{2} \sum_{i=1}^n w_i^2 \quad \text{u.d.N} \quad y_j(\mathbf{w}^T \mathbf{x}_j + b) \geq 1, 1 \leq j \leq N. \quad (4.4)$$

Dieses ist in geschlossener Form lösbar. Wenn die Klassen durch eine Hyperebene trennbar sind, kann somit immer das Optimum gefunden werden. Bei der Lösung des Optimierungsproblem ergibt sich, dass die Trennebene nicht durch alle Datenpunkte bestimmt wird, sondern nur durch die namensgebenden *Support Vektoren*. [STATNIKOV et al., 2011]

Das zuvor definierte quadratische Optimierungsproblem resultiert aus einen harten Margin, welches anfällig gegenüber Ausreißern ist. Deshalb wird meistens ein sogenannter *Soft-*

Margin verwendet, der die Nebenbedingung des Optimierungsproblem abschwächt, sodass Datenpunkte einer Klasse sich im Margin befinden können und sogar auf der anderen Seite der Trennebene. Aus einem Soft-Margin ergibt sich das Optimierungsproblem

$$\min_{\mathbf{w}} \frac{1}{2} \sum_{i=1}^n w_i^2 + C \sum_{i=1}^N \xi_i \quad \text{u.d.N} \quad y_j(\mathbf{w}^T \mathbf{x}_j + b) \geq 1 - \xi_j, 1 \leq j \leq N. \quad (4.5)$$

C ist ein Parameter zur Steuerung des Einflusses von Ausreißern. In Abbildung 4.2 sind die Support Vektoren und der Soft-Margin einer Gerade dargestellt.

Die SVM kann in der Grundform nur zwei Klassen voneinander unterscheiden. Diese müssen zudem durch eine lineare Hyperebene trennbar sein. Um das Problem von nicht linear trennbaren Klassen zu lösen, kann der sogenannte *Kernel-Trick* verwendet werden. Dadurch werden Datenpunkte in einem höher dimensional Raum transformiert, in denen die Datenpunkte linear trennbar sind. Für ein Mehrklassenproblem existieren unterschiedliche Ansätze. Ein möglicher Ansatz ist das Lernen von mehreren SVMs nach dem *One-vs.-One-* oder *One-vs.-Rest-*Schema.

4.3 Entscheidungsbaum

Dieser Abschnitt stellt den manuell erstellten Entscheidungsbaum vor und soll nur als Überblick dienen. Eine ausführliche Erläuterung des Entscheidungsbaumes findet sich im Anhang A.

Zur Übersicht sind in Abbildung 4.3 die Top-Level Ebenen des Entscheidungsbaumes dargestellt. Die Bedingungsknoten der ersten beiden Levels bilden die notwendige Bedingung für eine kritische Engstellensituation. Sie prüfen das Vorhandensein von Engstellen und Kollisionen sowie den Aufenthalt der Person und des Roboters im sogenannten Reaktionsbereich. Der Reaktionsbereich ist über die Entfernung der Person und des Roboters zur Engstelle definiert. Nur wenn sie sich darin befindet, soll eine Reaktion ausgeführt werden.

Wenn die notwendigen Bedingungen erfüllt sind, findet entsprechend der Bewegungsintention der Person eine Weiterleitung zu den entsprechenden Unterbäumen. Die Bewegungsintention bestimmt die Bewegungsrichtung der Person relativ zur Engstelle und unterscheidet zwischen stehenden, entgegenkommende (Gegenbewegung) und mitlaufenden Personen (Mitbewegung). Durch den verwendeten Ansatz existieren Fälle, in denen die Bewegungsintention nicht eindeutig bestimmbar ist. In diesen Fall soll keine Reaktion ausgeführt werden.

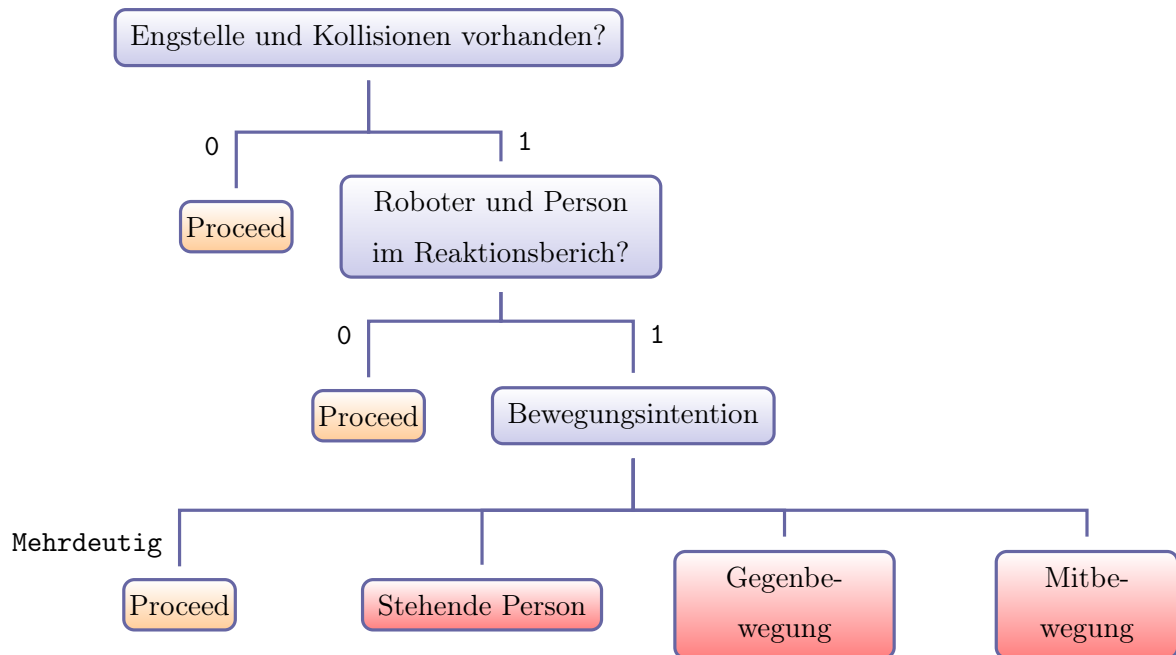


Abbildung 4.3: Darstellung der obersten Ebenen des entwickelten Entscheidungsbaumes. Die notwendigen Bedingungen für eine kritische Engstellensituation ist das Vorhandensein einer Engstelle und eines Platzkonfliktes, die mit Kollisionen prädiert werden. Zudem wird ein Reaktionsbereich um die Engstelle definiert. Wenn die notwendigen Bedingungen erfüllt sind, findet die weitere Verarbeitung entsprechend der dargestellten Bewegungsintention der Person statt. Im Unterbaum für stehende Personen werden Situationen mit Pfadblockierung ausgelöst durch in Engstellen stehende Personen. Die Unterbäume für die Gegenbewegung und Mitbewegung von Personen erkennen Parksituationen bzw. Einreihungssituationen.

Der Unterbaum zur Behandlung von stehenden Personen erkennt im Grunde, wann die Engstelle durch eine stehende Person blockiert ist. Im Fall einer Pfadblockierung wird der Merkmalsvektor zur Klasse Parking zugeordnet. Mit dem Unterbaum für entgegenkommende Personen werden Parksituationen erkannt. Dagegen erkennt der Unterbaum für mitlaufende Personen die Klasse der Einreihungssituationen. Auf einer groben Abstraktionsebene betrachtet, reagieren diese Unterbäume auf Kollisionen, die in einem kritischen Bereich um die Engstelle auftreten. Dieser kritische Bereich ähnelt den zuvor erwähnten Reaktionsbereich, ist jedoch über den Kollisionspunkt des Roboters definiert. Wenn eine Kollision im kritischen Bereich auftritt, werden über weitere Test über die Notwendigkeit einer Reaktion entschieden. Diese Test werden ausgehend von der Lage der Person und des Roboters: zur Engstelle, zur Kollision und zueinander ausgeführt.

Um die Situationen in die Klassen einzuteilen, werden eine Vielzahl von Bedingungen überprüft. Diese Überprüfung werden ausgehend von den extrahierten Merkmalsvektoren bestimmt. Dabei handelt es sich im Wesentlichen um logische Verknüpfungen der binären Ausgabe von Schwellwertoperationen. Diese Schwellwerte sind die Parameter des Entscheidungsbaums, über die die Klassenzuordnung gesteuert werden kann.

Kapitel 5

Experimentelle Untersuchungen

In diesem Kapitel werden die experimentellen Untersuchungen zur Bewertung des entwickelten Ansatzes beschrieben. Speziell wird die Performance der SVM und des Entscheidungsbaumes aus Kapitel 4 auf einen selbst erstellten Testdatensatz bestimmt und anschließend ausgewertet.

5.1 Datensatz

5.1.1 Aufnahmeverfahren

Zum Aufbau des Datensatzes wurden Aufnahmen genutzt, die während des Roboterbetriebes aufgezeichnet wurden. Die Aufnahmen können als eine Art *Videotape* verstanden werden. Sie enthalten die Low-Level Sensordaten als auch Daten von Modulen höherer Abstraktionsebenen und ermöglichen es den Roboterbetrieb zu einem späteren Zeitpunkt wiederzugeben.

Der Datensatz wurde aus Aufnahmen im Gebäude des Fachgebietes NIKR erstellt. Dazu wurden an verschiedenen Engstellen des Gebäudes eine Betriebsfahrt des Roboters mit möglichen Konflikten nachgestellt. Die so erstellten Aufnahmen wurden danach offline inspiziert und manuell gelabelt. Für die Vergabe der Labels wurden bei der Wiedergabe der Aufnahmen zu jedem Zeitpunkt die Merkmale extrahiert und visualisiert. Abhängig von diesen Visualisierungen wurden die Labels vergeben. Für jeden Zeitpunkt musste ein Label vergeben werden. Da sich die Merkmale oft blitzartig verändert haben und somit auch die Visualisierungen, stellte sich das Labeln als eine schwierige Tätigkeit heraus. In Abbildung 5.1 sind die Visualisierungen der Merkmale dargestellt. Die gelabelten Aufnahmen wurden anschließend zur Merkmalsextraktion und Klassenzuordnung genutzt.

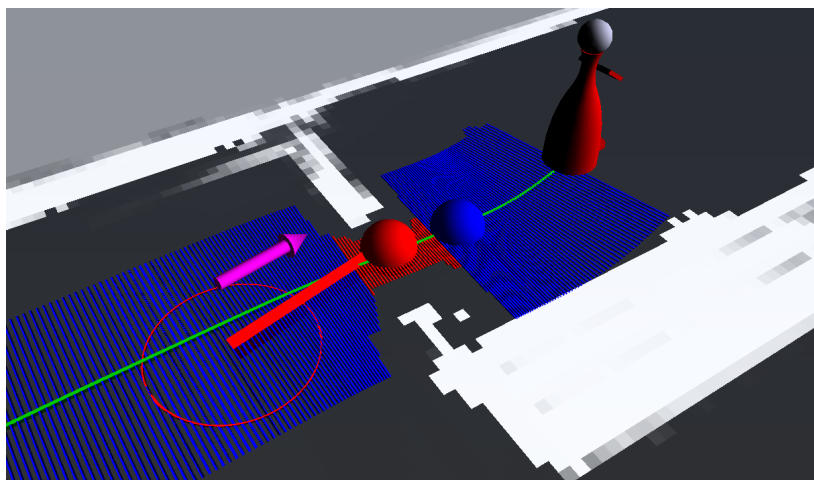


Abbildung 5.1: Ausschnitt einer Aufnahme mit den Visualisierungen der verwendeten Merkmale. Die grüne Linie stellt den geplanten Roboterpfad dar. Entlang dieser sind die Normalen durch blaue Geraden dargestellt. Normalen innerhalb von Engstellen sind rot markiert. Die Position von Personen wird durch einen roten Kreis dargestellt. Der magentafarbene Pfeil darüber ist der Geschwindigkeitsvektor der Person. Wenn es zu Kollisionen zwischen dem Roboter und einer Person kommt, werden die Kollisionspunkte von der Person bzw. dem Roboter durch gefärbte Kugeln dargestellt.

5.1.2 Aufnahmeumgebung

In den Gebäuden des Fachgebietes wurden Aufnahmen an drei verschiedenen Engstellen aufgezeichnet. In Abbildung 5.2 sind die Engstellen im Gebäudegrundriss markiert. Die Engstellen A und C sind statisch und resultieren aus der Gebäudestruktur. Die Engstelle B wurde dagegen durch künstliche Hindernisse erstellt. Für die Aufnahmen fuhr der Roboter entlang festgelegter Wegpunkte, die durch die Engstellen führten. Während dieser Fahrt bewegte sich eine Person in der Umgebung der Engstelle, um unterschiedliche konfliktäre Situationen hervorzurufen. Aus den Aufnahmen an den Engstellen wurden jeweils ein Datensatz erstellt. Insgesamt konnten circa 90 Minuten brauchbares Material aufgenommen werden. Daraus konnten für die definierten Klassen aus Abschnitt 2.1 Merkmalsvektoren extrahiert werden. In Tabelle 5.1b sind jeweils für die betrachteten Engstellen die Anzahl der Vektoren pro Klasse aufgelistet.



Abbildung 5.2: Gebäudegrundriss der Aufnahmeumgebung. Die rot markierten Bereiche sind die Engstellen der beschriebenen Aufnahmen. Die Engstelle A und C resultieren aus der Gebäudestruktur. Die Engstelle B wurde durch zwei künstliche Hindernisse erzeugt.

	A	B	C
Breite (m)	1.4	1.3	
Länge (m)	0.3	0.4	7.0

(a) Länge und Breite der Engstellen

	A	B	C
Parking	41	163	95
Queueing	17	47	16
Proceed	2743	654	282

(b) Anzahl der Merkmalsvektoren pro Klasse

Tabelle 5.1: (a) Kennzahlen zur Struktur der betrachteten Engstellen und (b) die Aufteilung der extrahierten Merkmalsvektoren in die betrachteten Klassen.

5.2 Modeltraining

Entscheidungsbaum

Für den entwickelten Entscheidungsbaum besteht das Training aus dem Anpassen der Parameter. Dazu wurde der Datensatz B benutzt. Als Gütemaß wurde die Gesamtklassifikationsrate verwendet. Die Parameter wurden dabei manuell angepasst, indem der Verlauf der falsch klassifizierten Merkmalsvektoren im Entscheidungsbaum untersucht wurde. Die Parameter wurden ausgehend von den Standardwerten in Anhang A.5 angepasst. In Tabelle 5.2 sind die veränderten Parameter zu erkennen. Im Grunde wurden die Parameter derart angepasst, dass der Roboter defensiver reagiert. Der auf dieser Weise bestimmte beste Parametersatz

Parameter	Standardwert	Angepasster Wert
a_{REB}	4 m	5.5 m
d_{REB}	4 m	5.5 m
a_{NE}	1 m	0.5 m
d_{VFR}	0.2 m	1.5 m

Tabelle 5.2: Die Anpassung der Parameter des Entscheidungsbaumes wurden ausgehend von den Standardwerten in Anhang A.5 ausgeführt. Im Wesentlichen resultieren die Parameter in ein defensiveres Verhalten.

erzielte eine Klassifikationsrate von 95.0 %.

Support Vektor Maschine

Für die Untersuchungen wurde eine lineare SVM mit Soft-Margin verwendet. Als Implementierung wurde die Bibliothek *LIBSVM* [CHANG und LIN, 2011] benutzt. Das Optimierungsproblem einer SVM mit Soft-Margin führt einen zusätzlichen Parameter C ein, der die Einfluss von Ausreißern steuert (siehe Abschnitt 4.2). Um den Parameterwert zu bestimmen, wurde die *Grid-Search-Funktion* der LIBSVM genutzt. Die Grid-Search erhält jeweils für die zu bestimmenden Parameter ein Intervallbereich als Eingabe. Diese Intervalle spannen einen Parameterraum auf. Dieser wird mit einem *Brute-Force-Ansatz* durchsucht, um die beste Kombination zu finden. Die Güte einer Kombination wird durch eine *k-fache Kreuzvalidierung* auf einem Trainingsdatensatz bestimmt. Als Maß wird die Gesamtklassifikationsrate benutzt.

Zum Training der linearen SVM wurde wie beim Entscheidungsbaum der Datensatz B verwendet. Die Parameter wurden mit einer 10-fachen Kreuzvalidierung bestimmt. Der Suchbereich von C wurde auf $C \in [-4, 4]$ begrenzt. Beim Grid-Search wurde die beste Klassifikationsrate von 86.22 % bei $C = 1$ erreicht.

5.3 Auswertung

Zum Modelltraining wurde der Datensatz B verwendet, da dieser im Gegensatz zu den anderen Datensätzen die einzelnen Klassen besser repräsentiert. Für die restlichen Datensätze A und C wurden jeweils eine Konfusionsmatrix mit den Richtig-Positiv-Raten (TPR) für die

		GT		
		Parking	Queueing	Proceed
PK	Parking	35	0	21
	Queueing	0	17	22
	Proceed	6	0	2700
<i>TPR</i>		85.4 %	100 %	98.4 %

(a) Datensatz A

		GT		
		Parking	Queueing	Proceed
PK	Parking	89	0	14
	Queueing	0	15	0
	Proceed	6	1	268
<i>TPR</i>		93.7 %	93.8 %	95.0 %

(b) Datensatz C

		GT		
		Parking	Queueing	Proceed
PK	Parking	145	0	12
	Queueing	0	45	11
	Proceed	18	2	631
<i>TPR</i>		89.0 %	95.7 %	96.5 %

(c) Trainingsdatensatz B

Tabelle 5.3: Konfusionsmatrizen des Entscheidungsbaumes.

		GT		
		Parking	Queueing	Proceed
PK	Parking	39	17	149
	Queueing	0	0	0
	Proceed	2	0	2594
<i>TPR</i>		95.1 %	0.0 %	94.6 %

(a) Datensatz A

		GT		
		Parking	Queueing	Proceed
PK	Parking	64	16	88
	Queueing	0	0	0
	Proceed	31	0	194
<i>TPR</i>		67.4 %	0.0 %	68.8 %

(b) Datensatz C

		GT		
		Parking	Queueing	Proceed
PK	Parking	143	43	52
	Queueing	0	0	0
	Proceed	20	4	602
<i>TPR</i>		87.8 %	0.0 %	92.0 %

(c) Trainingsdatensatz B

Tabelle 5.4: Konfusionsmatrizen der linearen SVM.

betrachteten Klassifikatoren bestimmt.

Es sei angemerkt, dass die Datensätze relativ klein und unausgeglichen sind. Die meisten Merkmalsvektoren stammen von der Klasse Proceed. Für die restlichen Klassen konnten nur wenige Merkmalsvektoren gewonnen werden. Für eine sichere Aussage über die Performance sollte der Datensatz erweitert werden, damit alle Klassen gleich gut repräsentiert werden. Trotz dessen kann die folgende Auswertung als ein Ausgangspunkt für weiterführende Arbeiten genutzt werden.

Entscheidungsbaum

In Tabelle 5.3 sind die Konfusionsmatrizen des Entscheidungsbaumes dargestellt. Der Entscheidungsbaum erreichte auf den beiden Datensätze eine relativ gute Performance. Außer für die Klasse Parking im Datensatz A erzielte die TPR immer einen Wert von mindestens 90.0 %. Im Durchschnitt wird eine TPR von 94.3 % erreicht. Folglich werden die Situationen der betrachteten Datensätze mit dem Entscheidungsbaum im Durchschnitt korrekt erkannt.

Jedoch traten auch Fehldektionen auf. Die meisten Fehldetektionen rief die Klasse Proceed auf. Diese Klasse repräsentiert alle Situationen, die keine Behandlung benötigen. Die Verwechslungen traten dabei mit den Klassen Parking und Queueing auf, welche Situationen mit einer Behandlung repräsentiert. Bei den Klassen Parking und Queueing traten Fehldektionen dagegen seltener auf. Aus den Konfusionsmatrizen kann zudem geschlossen werden, dass die Verwechslung nur mit der Klasse Proceed stattfand. Aus zeitlichen Gründen konnte dieser Sachverhalt nicht genauer untersucht werden. Es wird vermutet, dass die Ursache eventuell im verwendeten Verfahren zum Label der Aufnahmen liegt. Die Labels wurden während der Wiedergabe der Aufnahmen ausgehend von den Visualisierungen der Merkmale vergeben. Weil diese sich oft blitzartig verändert haben, kann es passiert sein, dass bei diesen Übergang Zeitpunkte ein nicht beabsichtigtes Label erhalten.

Support Vektor Maschine

Die Konfusionsmatrizen der linearen SVM sind in Tabelle 5.4 dargestellt. Für den Datensatz A kann die SVM die Klassen Proceed und Parking mit großer Sicherheit erkennen. Diese beiden Klassen erreichen eine TPR von mindestens 90.0 %. Allerdings kann die Klasse Queueing nicht korrekt erkannt werden. Jede Instanz von Queueing wird mit der Klasse Parking verwechselt. Für den Datensatz C erzielt die Klasse Proceed eine TPR von 68.8 %. Die Klasse Parking

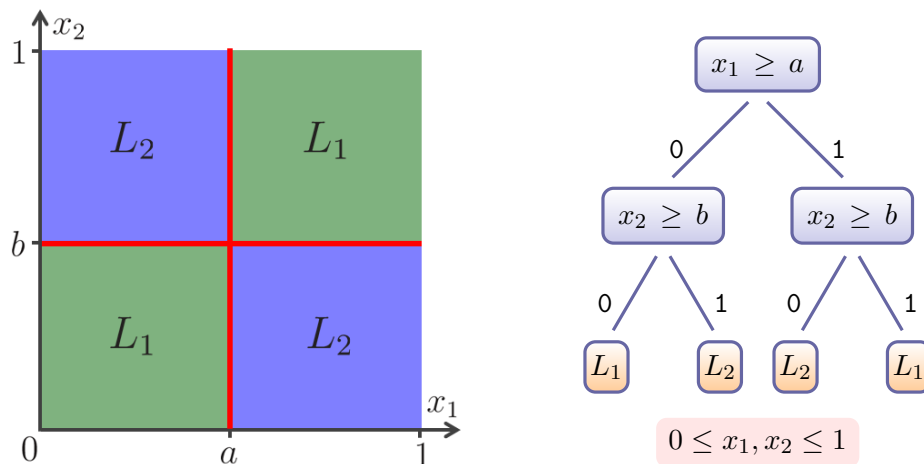


Abbildung 5.3: Nichtlineare Trenngerade eines Entscheidungsbaumes zum Lösen des XOR-Problems. Die Trenngerade wird durch die rote Linie dargestellt.

erreicht einen Wert von 67.4 %. Auch hier kommt es zu einer vollständigen Verwechslung der Queueing-Klasse mit der Parking-Klasse.

Der Umgebungsstruktur der Engstellen des Datensatzes A und des Trainingsdatensatzes B ähneln sich (siehe Abbildung 5.2). Beide Engstellen haben eine Länge von circa 0.3 m. Dadurch wird vermutet, dass die hervorgerufenen Situationen in diesen Datensätzen sich ebenfalls ähneln. Im Gegensatz dazu unterscheidet sich die Umgebungsstruktur der Engstelle aus dem Datensatz C stark vom Trainingsdatensatz und hat zudem mit einer Länge von 7 m eine wesentlich größere Ausdehnung. Es wird daher vermutet, dass die Ähnlichkeit zwischen dem Datensatz A und dem Trainingsdatensatz in den guten Erkennungsraten der Klassen Proceed und Parking aus Tabelle 5.4a resultieren. Während die Erkennungsraten für diese Klassen im Datensatz C eine deutliche Verschlechterung aufweisen. Durch den Unterschied der Engstellenart sind im Datensatz C Situationen enthalten, die stark vom Trainingsdatensatz abweichen. Die SVM kann dann diese Situationen nicht erkennen, da für diese keine Daten vorlagen.

Die Queuing-Klasse wird in den Testdatensätzen vollständig mit der Parking-Klasse verwechselt. Aufgrund der guten Trennbarkeit dieser Klassen mit dem Entscheidungsbaum wird vermutet, dass diese beiden Klassen nicht linear voneinander trennbar sind. Diese Vermutung wird auch von der Konfusionsmatrix der linearen SVM auf dem Trainingsdatensatz bestärkt (siehe Tabelle 5.4c). Auch auf Trainingsdatensatz findet die Verwechslung statt, sodass eine Überspezialisierung mit großer Sicherheit ausgeschlossen werden kann.

Ein Entscheidungsbaum kann als mehrschichtiges System betrachtet werden. Er besteht aus mehreren Knoten, die schichtenweise angeordnet sind. Die Blätter des Baumes repräsentieren dabei die endgültigen Entscheidungen. Auf dem Pfad zu den Blättern werden in jedem inneren Knoten Zwischenentscheidungen getroffen. Der entwickelte Entscheidungsbaum dieser Arbeit nutzt in den inneren Knoten ausschließlich Schwellwertoperationen, die über logische Operatoren verknüpft werden. Die Schwellwertoperationen werden auf einzelne Merkmale des Merkmalsvektor angewendet. Wenn die vereinfachende Bedingung angenommen wird, dass jeder Knoten nur eine Schwellwertoperation auf ein Merkmal anwendet, findet in jedem dieser Knoten eine Trennung des Merkmalsraumes durch eine Hyperebene statt. Diese Trennung verschachtelt sich über die fortschreitende Tiefe des Entscheidungsbaumes. Die resultierende Hyperebene wird somit nichtlinear. In Abbildung 5.3 ist ein Beispiel einer nichtlinearen Trenngerade in einem zweidimensionalen Merkmalsraum dargestellt. Weitere Untersuchungen sollten daher eine nichtlineare SVM verwenden.

Die gute Erkennungsrate des Entscheidungsbaumes kann auf die verwendeten Merkmale zurückgeführt werden. Diese Merkmale haben einen relativ hohen Abstraktionsgrad und sind dadurch gut interpretierbar, sodass ein Design des Entscheidungsbaumes per Hand möglich ist. Dagegen benötigt die SVM Trainingssamples für alle zu erkennenden Situationen. Mit einem größeren Datensatz, der die Situationen besser repräsentiert, würde auch die Erkennungsrate der SVM einen guten Wert erreichen.

Kapitel 6

Zusammenfassung und Ausblick

6.1 Zusammenfassung

Die vorliegende Arbeit wurde im Rahmen des Projektes ROREAS ausgeführt. Ziel von ROREAS ist die Entwicklung eines robotischen Laufassistenten für Schlaganfallpatienten. Der Roboter soll dabei den Patienten bei seinem eigenständigen Lauftraining unterstützend begleiten und das Training auf abrechenbarer Weise aufzeichnen. Für einen robotischen Assistenten ist eine höfliche und nutzerzentrierte Navigation ein wichtiger Bestandteil für die soziale Akzeptanz in seiner Einsatzumgebung.

Der konkrete Einsatzort ist die Fachklinik in Bad Liebenstein. Dort wird das Lauftraining hauptsächlich auf den Korridoren ausgeführt. Aufgrund der Gebäudestruktur und den abgestellten Arbeitsgeräten ist der seitliche Platz in manchen Bereichen stark begrenzt, sodass Engstellen entstehen können. Diese Engstellen geben einen Bewegungsfluss vor, der gleichzeitig nur eine Bewegung in eine Richtung zulässt. Wenn sich der Roboter und eine Person sich gleichzeitig durch eine Engstelle bewegen möchte, können Verklemmungssituationen entstehen. Zur Gewährleistung einer höflichen Navigation müssen Verklemmungen im Voraus erkannt werden und geeignet behandelt werden.

Im entwickeltem Ansatz dieser Arbeit werden Situationen durch einen reellwertigen Merkmalsvektor beschrieben. Die Merkmale werden vom geplanten Roboterpfad und den Bewegungszustand der detektierten Personen sowie der Belegheitskarte extrahiert. Die Merkmale beschreiben Platzkonflikte des Roboters mit den Personen in Zusammenhang mit den detektierten Engstellen. Im Wesentlichen enthalten die Merkmale die raumzeitliche Lage des Roboters und einer Person mit Platzkonflikten zu einer detektierten Engstelle.

Die Platzkonflikte werden durch Kollisionen der Person mit dem Roboterpfad beschrieben. Durch Nutzung eines linearen Bewegungsmodell können die Platzkonflikte prädiziert werden und ermöglichen somit die Erkennung von Verklemmungssituationen im Voraus. Zur Prädiktion von Kollisionen werden die raumzeitlichen Trajektorien des Roboters und der Person bestimmt. Unter der Annahme einer Kreisform für den Roboter und der Person sind Kollisionen Zeitpunkte auf den Trajektorien, in denen sich diese Kreisformen überschneiden.

Für die Erkennung von Engstellen werden für diskrete Punkte auf den geplanten Roboterpfad Normalen berechnet, die senkrecht zum Pfad verlaufen. Entlang dieser Normalen wird ein Raycasting in der Belegheitskarte ausgeführt, um die Distanz zu Hindernissen zu bestimmen. Diese können als Durchfahrtsbreiten betrachtet werden. Engstellen sind dann zusammenhängende Pfadabschnitte, in denen die maximale Durchfahrtsbreite nur Platz für den Roboter oder eine Person bietet.

Jede Situation wird auf einen Merkmalsvektor reduziert, der einen Punkt im mehrdimensionalen Raum ist. In diesen Raum werden Verklemmungssituationen mit Hilfe eines Klassifikators erkannt. Dazu werden die Merkmalsvektoren zu Klassen zusammengefasst, welche die Behandlung ihrer entsprechenden Situation repräsentiert. Der Klassifikator bildet also jede Situation auf die entsprechende Behandlung ab. Für die Engstellen der Klinikumgebung können die Situationen in die drei Klassen: Unkritische Situationen, Parksituationen und Einreihungssituationen eingeteilt werden. Als Klassifikatoren wurde eine lineare SVM und ein manuell erstellter Entscheidungsbaum genutzt.

Für die experimentellen Untersuchungen wurde ein Datensatz im Gebäude des Fachgebietes NIKR aufgenommen. Der Datensatz besteht aus drei Engstellen mit einer Länge von 0.3 m (A) 0.4 m (B) und 7.0 m (C). Der Datensatz B wurde zum Training der Klassifikatoren genutzt. Auf den Datensätzen A und C wurden die verwendeten Klassifikatoren evaluiert. Der Entscheidungsbaum schnitt mit relativ guten Erkennungsraten ab. Der Durchschnitt der TPR über die drei Klassen und den zwei Testdatensätzen beträgt dabei 94.3 %. Die lineare SVM konnte die unkritischen Situationen und die Parksituationen in den Datensatz A mit einer TPR von 94.1 % bzw. 95.1 % erkennen. Im Datensatz C sanken diese auf 68.8 % bzw. 67.4 %. Es wird vermutet, dass der Datensatz C Situationen enthält, die nicht im Datensatz A vorhanden sind. Die SVM kann dann diese Situationen, da keine Trainingsamples vorlagen. Die Einreihungssituationen konnte von der linearen SVM nicht erkannt werden. Jede Instanz der Einreihungssituation wurde dabei immer als eine Parksituation erkannt. Es wird vermutet, dass dies in der nicht linearen Trennbarkeit der Einreihungssituationen

und der Parksituationen liegt. Die Untersuchungen wurden auf einen relativ kleinen und unausgeglichenen Datensatz ausgeführt. Für eine sichere Aussage zur Performance sollte ein größerer Datensatz benutzt werden.

6.2 Ausblick

Aufgrund des zeitlich begrenzten Rahmen dieser Arbeit konnte nur der merkmalsbasierte Klassifikationsansatz zur Realisierung der Situationserkennung verfolgt werden. Daher kann die vorliegende Arbeit als eine erste Version betrachtet werden, die es dennoch ermöglicht einen tieferen Einblick in die Thematik zu erlangen. Aus diesen gewonnenen Erkenntnissen resultieren die nachfolgenden Verbesserungsvorschläge für das entwickelte Verfahren sowie eine Diskussion über neue Ansätze.

Die Evaluationen auf den aufgenommenen Testdatensatz zeigen, dass die Situationserkennung in den getesteten Engstellen gute Erkennungsraten erzielt. Bei Test in realen Umgebungen wurde festgestellt, dass die Erkennungsraten stark von Eingaben des Verfahren abhängt. Wenn z.B. Fehldetektionen beim Personentracker entstehen, kann dies zu einer Fehlentscheidung der Situationserkennung führen. Neben den nachfolgenden Verbesserungsvorschlägen würde die Situationserkennung deshalb auch von den Verbesserungen der externen Abhängigkeiten profitieren.

6.2.1 Datenerfassung

Labeling vom Datensatz

Zum Training und zur Evaluation der SVM wurde ein Datensatz im Gebäude des Fachgebietes NIKR aufgenommen. Einige Situationen wurden dabei zu defensiv gelabelt. Dies ergab sich aus Tests in der realen Umgebung. Die Erkennung mit der SVM löst in machen Situationen bei der Detektion von Kollisionen in der Nähe der Engstelle ein vorzeitiges Beiseitefahren des Roboters aus, obwohl eine Weiterfahrt den Kollisionspartner nur minimal oder gar nicht beeinträchtigt. Die Label dieser Situationen sollten für eine flüssigere Bewegung des Roboters nachträglich verändert werden.

Erweiterung des Datensatz mit neuen Engstellen

Der derzeitig verwendete Datensatz besteht aus Aufnahmen von drei Engstellen. Zwei dieser Engstellen besitzen eine relativ kurze Ausdehnung. Sie haben eine Länge von 0,3 m und 0,4 m. Die dritte Engstelle wurde auf einen Korridor aufgenommen und hat eine Länge von circa 7 m.

Wenn die SVM nur mit den Daten der kurzen Engstelle trainiert wurde, schnitt sie auf den Daten des langen Korridors mit einer niedrigen Erkennungsrate ab. Es wird vermutet, dass der Korridor Situationen hervorruft, die stark von denen der kurzen Engstellen abweicht. Dadurch kann die SVM nur schlecht generalisieren. Dabei hat voraussichtlich nicht nur die Engstellenlänge einen beeinflussenden Faktor. Um die Annahme zu überprüfen, sollte das Verfahren auf weiteren Engstellenarten getestet werden. Wenn die Annahme sich als richtig erweist, ist es erforderlich den Datensatz mit weiteren Engstellenarten zu erweitern.

6.2.2 Patientenbezogene Merkmale

In ROREAS muss der Roboter neben normalen Betriebsfahrten, die der Roboter allein ausführt, einen Patienten unterstützend begleiten. Die verwendeten Merkmale dieser Arbeit berücksichtigen allerdings den begleiteten Patienten nicht. Durch die Zunahme des Patienten können neue Situationen entstehen, die nicht von den jetzigen Merkmalsraum abgedeckt werden. Eine mögliche Erweiterung des Merkmalsraum wäre die Hinzunahme der Distanz des Patienten zum Roboter und dem Winkel des Aufenthaltspunktes des Patienten relativ zur aktuellen Bewegungsrichtung des Roboters. Damit würde die relative Lage des Patienten zum Roboter erfasst werden.

6.2.3 Prädiktive Kollisionserkennung

Berücksichtigung der Umgebungskarte

Die Situationserkennung ist für typische Engstellen auf langen Korridoren ausgerichtet. Diese Engstellen trennen den Korridor in zwei Bereiche. Um von einem Bereich in den Nächsten zu gelangen, gibt es keinen effizienteren Weg als durch die Engstelle. Die Engstelle gibt somit ein Bewegungsfluss vor, der parallel zu den umschließenden Wänden ist. Das bedeutet auch, dass die Engstelle seitlich abgeschlossen ist. Personen können sich nur entlang des vorgebenen Bewegungsflusses bewegen. Eine Annäherung an den Engstellenseiten wird damit ausgeschlossen.

Die derzeitige prädiktive Kollisionserkennung geht von einem linearen Bewegungsmodell der Personen und des Roboters aus. Personen werden dabei gleichförmig entlang ihres Geschwindigkeitsvektor bewegt. Die prädiktive Bewegung der Personen geschieht dabei ohne Informationen aus der Umgebungskarte. Wenn es möglich ist, dass Personen sich an den Engstellenseiten annähern, können dadurch hervorgerufene Kollisionen fälschlicherweise als Verklemmungen erkannt werden.

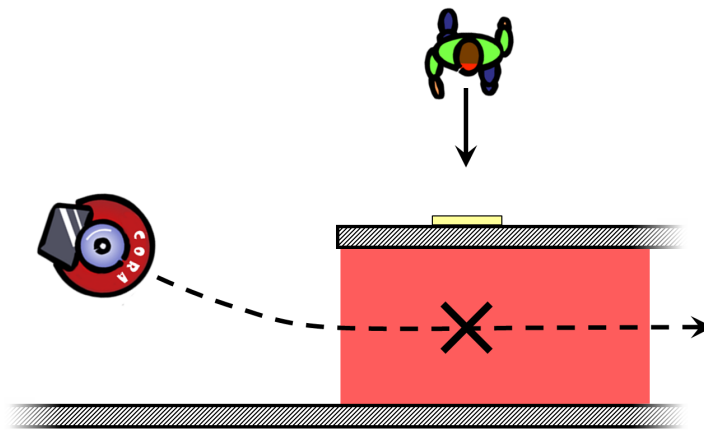


Abbildung 6.1: Illustration einer als fälschlicherweise erkannten Verklemmungssituation unter Nichtbeachtung der Umgebungskarte. Der Roboter möchte sich durch die Engstelle bewegen und detektiert eine Person, die sich auf die Engstellenseite bewegt. Weil die Umgebungskarte nicht berücksichtigt wird, wird eine unmögliche Bewegung durch die Engstellenseite prädiziert. Es kommt dadurch zur Kollision mit dem Roboter in der Engstelle, die als Konfliktsituation erkannt wird.

In Abbildung 6.1 ist ein Beispiel einer solchen Situation dargestellt. Weil die Informationen der Karte nicht mitberücksichtigt wird, bewegt sich die Person durch die Engstellenwände und es kommt zu einer nicht möglichen Kollision in der Engstelle. Damit solche Situationen nicht eine unnötige Unterbrechung des Roboters hervorrufen, sollte die Informationen der Umgebungskarte bei der Prädiktion mitberücksichtigt werden.

Maskierung von Kollisionen

Die verwendeten Support Vektor Maschine der prädiktiven Kollisionserkennung beziehen sich nur auf die Person mit der zeitlich nächsten Kollision. Dadurch werden alle Personen mit einer späteren Kollision bei der Erkennung nicht berücksichtigt. Es können jedoch Situationen entstehen, in denen die zeitlich nächste Kollision unproblematisch ist und dadurch die späte-

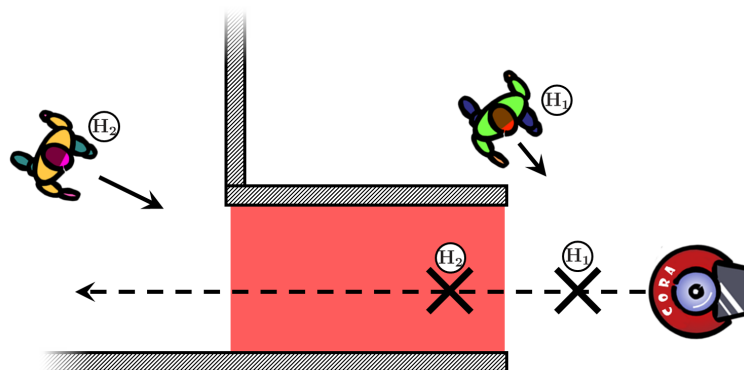


Abbildung 6.2: Darstellung eines Beispiels einer problematischen Maskierung einer kollisionsgefährlichen Kollision. Der geplante Pfad des Roboters führt durch die Engstelle und es werden Kollisionen mit den Personen H_1 und H_2 detektiert. Weil bei der Situationserkennung nur die zeitlich nächste Kollision betrachtet wird, maskiert die Kollision mit H_1 die zeitlich nachfolgende kollisionsgefährliche Kollision mit H_2 . Erst wenn die Kollision mit H_1 passiert ist und der Roboter direkt vor der Engstelle steht, wird die kollisionsgefährliche Kollision mit H_2 erkannt.

ren Kollisionen maskiert. Die Erkennung der nachfolgenden Kollisionen verzögert sich bis die zeitlich nächste Kollision passiert ist. Je nach dem Konfliktpotential und der zeitlichen Nähe der nachfolgenden Kollisionen kann eine Verklemmungssituation dadurch zu spät erkannt werden.

In Abbildung 6.2 ist ein Beispiel einer problematischen Maskierung illustriert. Die Person H_1 erzeugt nahe dem Engstellenbeginn eine unproblematische Kollision, welche die verklemmungsauslösende Kollision mit H_2 maskiert. Erst wenn der Kollisionsschwerpunkt mit H_1 passiert ist und der Roboter vor der Verklemmung steht, wird die Kollision mit H_2 erkannt.

Zur Vermeidung einer solchen Maskierung muss die Situationserkennung alle Kollisionen betrachten. Dazu sollten die Kollisionen zeitlich fortschreitend abgearbeitet werden bis eine kollisionsgefährliche Kollision gefunden wird oder alle Kollisionen abgearbeitet sind.

Verbesserung der Intentionsschätzung

Nicht alle prädizierten Kollisionen erzeugen in Präsenz einer Engstelle eine Verklemmungssituation. Die entscheidenden Faktoren dafür sind die Nähe des Kollisionsschwerpunktes zur Engstelle und die Intention des Kollisionsschwerpunktes sich durch die Engstelle zu bewegen. Derzeit wird diese Intention nur aus dem Winkel der Bewegungsrichtung zum Start bzw. Ende der Engstelle geschlossen. Für manche Situationen ist dies nicht ausreichend.

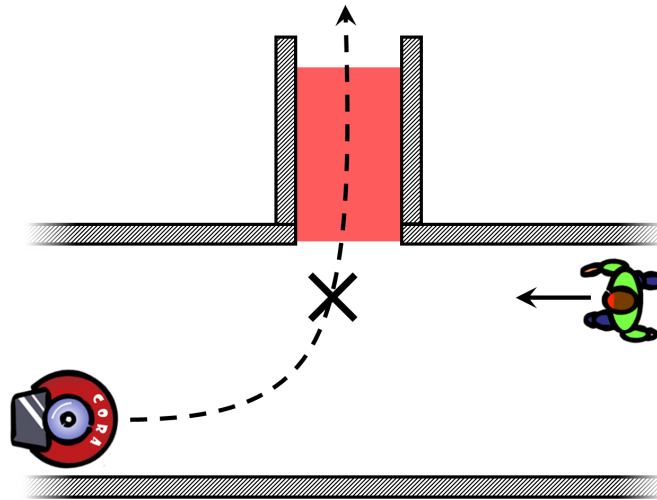


Abbildung 6.3: Die Bewegungsintention der entgegenkommenden Person kann in dieser Situation nicht aus dem Winkel der Bewegungsrichtung zum Start der Engstelle bestimmt werden. Ohne die Bewegungsintention ist es den Roboter nicht möglich die Situation richtig einzuschätzen. Die Person könnte sich durch die Engstelle bewegen und eine Verklemmung hervorrufen. Sie könnte sich aber auch an der Engstelle vorbei bewegen, wodurch keine Behandlung der Situation nötig ist.

In Abbildung 6.3 ist eine solche Situation dargestellt. Der Roboter möchte sich durch die Engstelle bewegen und es kommt zu einer Kollision mit einer entgegenkommenden Person. Diese Person bewegt sich senkrecht auf dem Engstellenbeginn zu. In diesem Fall könnte sich die Person durch die Engstelle bewegen oder auch daran vorbei. Nur mit Hilfe des Winkels der Bewegungsrichtung ist die sichere Bestimmung der Intention nicht möglich.

Eine mögliche Lösung wäre es in solchen Situationen immer ein Beiseitefahren des Roboters auszulösen und zu warten bis die Person die Engstelle passiert hat. Dies ist eine akzeptable Lösung für Einsatzumgebungen, in denen solche Situationen nur selten auftreten. Wenn allerdings diese Situationen häufig auftreten, hindert dies ein Vorankommen des Roboters und es sollte die Erfassung der Bewegungsintention verbessert werden. Es könnte z.B. der zeitlich Verlauf der Personenbewegung berücksichtigt werden.

6.2.4 Berücksichtigung der Personenbewegung in der Engstellenerkennung

Zur Erkennung von Engstellen werden für diskrete Punkte des geplanten Pfades Normalen berechnet. Entlang der Normalen wird ein Strahl in der Navigationskarte erzeugt, um die Durchfahrtsbreite zu bestimmen. In der Navigationskarte sind die Informationen der globalen Karte mit den statischen und dynamischen Hindernissen der lokalen Karte vereint.

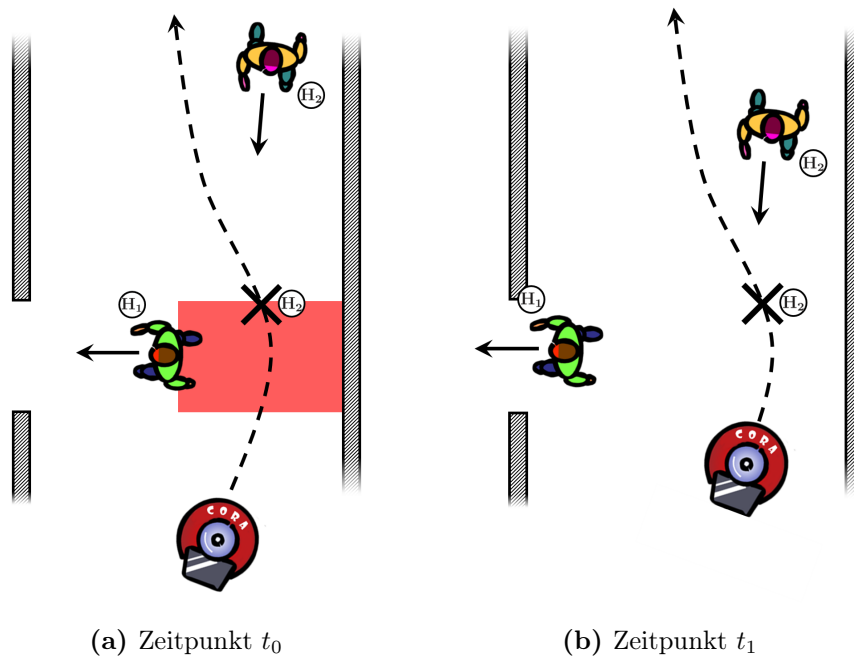


Abbildung 6.4: Darstellung eines Beispiel einer fälschlichen Verklemmungssituation unter Nichtbeachtung der Personenbewegung. Für den Zeitpunkt t_0 plant der Roboter den dargestellten Pfad um die Person H_1 . Der Pfadabschnitt um die Person wird als Engstelle erkannt. Zum Zeitpunkt t_1 existiert die Engstelle nicht, da sich H_1 bewegt hat. Wird noch zusätzlich zum Zeitpunkt t_0 eine Kollision mit einer entgegenkommenden Person H_2 in der Engstelle erkannt, so wird eine unnötige Unterbrechung des Roboters ausgelöst.

In der verwendeten Navigationskarte ist es nicht möglich Personen von statischen Hindernissen zu unterscheiden. Wenn der Roboter einen Pfad entlang einer sich bewegenden Person plant, können fälschlicherweise Engstellen erkannt werden. Ein solcher Fall ist in Abbildung 6.4 dargestellt. Dort wird im aktuellem Zeitpunkt eine Engstelle erkannt, die durch eine Person verursacht wird. Durch die seitliche Bewegung dieser Person existiert die Engstelle für zukünftige Zeitpunkte nicht mehr. Weil die Bewegung nicht mitberücksichtigt wird, werden

Kollisionen in der Nähe der Engstelle fälschlicherweise als Verklemmung erkannt.

Damit solche Situationen zu keinen Fehlentscheidungen führen, sollte die Engstellenerkennung um eine zeitliche Dimension erweitert werden. Dazu ist es notwendig, dass in der Navigationskarte bewegliche Hindernisse von statischen Hindernissen unterscheidbar sind.

6.2.5 Weitere Ansätze

Zur Realisierung der Situationserkennung wird in dieser Arbeit ein expliziter Klassifikationsansatz verfolgt. Es wird versucht die Situationen explizit mit Hilfe von Merkmalen zu beschreiben. Eine Situation kann dadurch als ein Punkt in einen mehrdimensionalen Merkmalsraum repräsentiert werden. Entsprechend der gewünschten Aktion in den einzelnen Situationen werden die Merkmalsvektoren in Klassen aufgeteilt. Ein Klassifikator würde somit jede Situation auf eine situationsgerechte Aktion abbilden. Als Klassifikatoren werden in dieser Arbeit die SVM als lernbasiertes Verfahren und ein manuell erstellter Entscheidungsbaum genutzt.

Zum Training der SVM mussten Daten in realer Umgebung aufgenommen und anschließend gelabelt werden. Dieser Prozess erwies sich als sehr zeitintensiv. Denn damit die SVM im realen Betrieb gute Erkennungsraten erzielt, müssen unterschiedliche Situationen nachgestellt werden. Wie in Abschnitt 6.2.1 (Erweiterung des Datensatz mit neuen Engstellen) bereits erwähnt, hat dies einen wichtigen Einfluss auf die Generalisierungsfähigkeit der SVM. Gegenüber der SVM ist die Erstellung des Entscheidungsbaumes weniger zeitintensiv, weist aber ebenfalls die selben Schwierigkeiten auf. Wenn eine Situation im Entscheidungsbaum nicht abgedeckt wird, muss der Entscheidungsbaum im schlechtesten Fall neu aufgebaut werden.

Ein Nachteil des verfolgtem Ansatz dieser Arbeit ist, dass Situationen explizit erfasst und explizit bewertet werden müssen. Je nach Komplexität der Situation ist es schwierig jede relevante Situation zu kennen, um diese in einem Datensatz aufnehmen zu können oder im voraus zu planen. Dadurch besteht die Gefahr, dass Situationen nicht abgedeckt werden.

Folglich wäre es besser die Situation nur implizit zu erfassen und auf eine explizite Beschreibung zu verzichten. Dies könnte durch eine raumzeitliche Pfadplanung realisiert werden, die es erlaubt die Bewegung von dynamischen Hindernissen zu berücksichtigen. Wenn der Roboter und eine Person sich gleichzeitig durch eine Engstelle bewegen, würde dies in die Pfadplanung einfließen. Der Planer könnte aus diesen Informationen einen nicht verklemmenden

raumzeitlichen Pfad herleiten. Die Nutzung eines raumzeitlichen Planers könnte allerdings den Einsatz in der realen Umgebung behindern, da durch die Erweiterung der Planung um eine zeitliche Komponente der Suchraum eine weitere Dimension erhält, welches die Laufzeit negativ beeinträchtigen könnte.

Ein weiterer Ansatz wäre Reinforcement Learning. Der Roboter könnte dabei autonom in einer Umgebung ein Verhalten bei Engstellen lernen, welches über die Reward Funktion implizit vorgegeben wird. Gegenüber einem überwachten Lernverfahren entfällt die manuelle Aufnahme und das Labeling eines Datensatzes. Je nach den auftretenden Engstellen in der Lernumgebung könnten dadurch allerdings Situationen nicht abgedeckt werden.

6.2.6 Parkposition

Die Bestimmung der Parkposition hat einen nicht zu vernachlässigen Einfluss auf die Situationserkennung, welches Auswirkungen auf den Gesamtablauf der Engstellenbehandlung als auch auf die Erkennungsrate hat. Für den Gesamtablauf sollte eine gut gewählte Parkposition, den Roboter eine schnelle und reibungslose Unterbrechung und Wiederaufnahme seines aktuellen Verhalten ermöglichen, ohne seine Umgebung stark zu beeinträchtigen. Für eine gute Erkennungsrate sollte zudem eine Parkposition auch einen guten Beobachtungspunkt auf die Engstelle liefern. Nur damit kann die Situationserkennung sichere Entscheidungen treffen, um den Roboter aus der Parkposition freizugeben.

Derzeit wird die vorhandene Bestimmung der Parkposition von vorangehenden Arbeiten des Fachgebietes benutzt. Sie besteht ähnlich dem Dynamic Window Approach aus mehreren Objectives, die für jede Position in einer festgelegten Umgebung des Roboters einen Wert für die Eignung als Parkposition vergeben. Die Position mit dem höchsten Eignungswert wird als Parkposition gewählt.

In den Test wurde festgestellt, dass die Parkpositionsbestimmung ein noch nicht ausgeschöpftes Verbesserungspotential hat. In manchen Fällen wird eine nicht optimale Position gewählt. Diese ist zu weit entfernt oder liegt in einem Bereich der nicht vom Roboter erreicht werden kann. Des Weiteren wurde festgestellt, dass die Bestimmung der Parkposition nur per Anfrage berechnet wird. Auf der Fahrt zur Position findet keine Neuberechnung statt, auch wenn die Navigationskarte sich geändert hat. Nur durch eine ständige Anfrage kann eine Änderung der Karte berücksichtigt werden. Dies ist jedoch ineffizient und sollte deshalb intern von der Parkpositionsbestimmung realisiert werden.

Aus zeitlichen Gründen konnte während der Arbeit keine Untersuchung für die nicht optimale Wahl der Parkposition durchgeführt werden. Es wird vermutet, dass die Ursachen nicht im Verfahren, sondern in der Implementierung liegen. Es kann zudem nicht ausgeschlossen werden, dass dies auch ein Problem der Konfiguration sein könnte.

Anhang A

Entscheidungsbaum

Eine Übersicht zu den Top-Level Ebenen findet sich im Abschnitt 4.3. Diese bilden die notwendigen Bedingungen für die Einordnung in die Klassen Parking und Queueing. Im folgenden werden die Unterbäume nach der Unterscheidung der Bewegungsintention erläutert. Dazu werden zunächst im folgenden Abschnitt wichtige Bedingungen eingeführt, die wiederkehrend im Entscheidungsbaum genutzt werden.

A.1 Entscheidungsbedingungen

Im folgenden werden Bedingungen erläutert die wiederkehrend im Entscheidungsbaum genutzt werden. Zudem wird auf die Bestimmung der Bewegungsintention und des Reaktionsbereiches der Top-Level Ebenen eingegangen. Die verwendeten Bezeichnungen der Merkmale beziehen sich auf die Abbildungen 3.2 und 3.4. Für die weiteren Betrachtungen wird angenommen, dass die Winkel aus den Merkmalen über das Skalarprodukt von zwei Vektoren bestimmt werden. Zum Beispiel wird der Winkel δ_{MS} aus den Merkmalen über die Bewegungsrichtung der Person und die Bewegungsrichtung des Roboters am Engstellenstart gebildet. Für den Winkel zwischen zwei Vektoren \mathbf{a} und \mathbf{b} gilt dann:

$$\angle(\mathbf{a}, \mathbf{b}) = \arccos\left(\frac{\mathbf{a} \circ \mathbf{b}}{\|\mathbf{a}\| \|\mathbf{b}\|}\right). \quad (\text{A.1})$$

Alle im folgenden betrachteten Winkel nehmen folglich einen Wert in $[0, \pi]$ an.

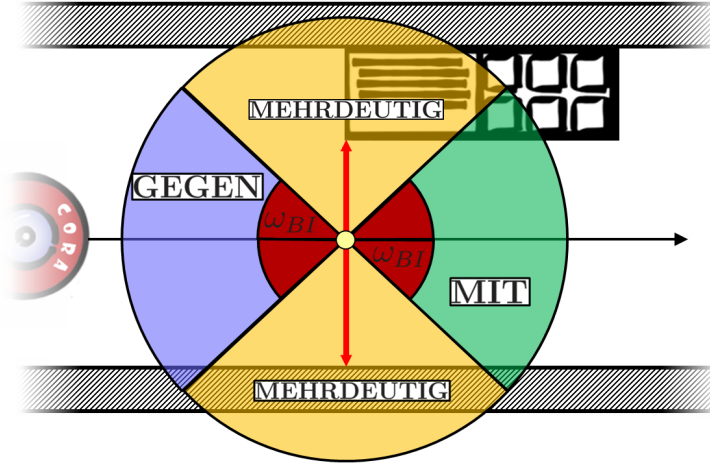


Abbildung A.1: Die durch ω_{BI} definierten Winkelbereiche zur Bestimmung der Bewegungsintention einer Person. Zur Bestimmung der Bewegungsintention wird der Winkel δ_{MS} zwischen der Bewegungsrichtung einer Person und der Bewegungsrichtung des Roboters am Engstellenstart verwendet.

A.1.1 Bewegungsintention

$$BI(v_h, \delta_{MS} | v_{min}, \omega_{BI}) = \begin{cases} STEHEND, & v_h < v_{min} \\ MIT, & v_h \geq v_{min} \wedge \delta_{MS} \leq \omega_{BI} \\ GEGEN, & v_h \geq v_{min} \wedge \pi - \delta_{MS} \leq \omega_{BI} \\ MEHRDEUTIG, & \text{sonst} \end{cases} \quad (A.2)$$

- v_h : Geschwindigkeit der Person.
- δ_{MS} : Winkel der Bewegungsrichtung der Person relativ zur Bewegungsrichtung des Roboters am Engstellenstart.
- v_{min} : Schwellwert der minimalen Geschwindigkeit einer sich bewegenden Person.
- ω_{BI} : Winkel zur Bestimmung der Bewegungsrichtung der Person.

Es wird zwischen den Zuständen $Z_{BI} = \{STEHEND, MIT, GEGEN, MEHRDEUTIG\}$ unterschieden. In Abbildung A.1 sind die durch ω_{BI} definierten Bereiche schematisch dargestellt.

A.1.2 Relative Lage der Person zum Roboter

$$P_{Roboter, Person}(\delta_{RH} | \omega_{PRP}) = \begin{cases} VOR, & \delta_{RH} \leq \omega_{PRP} \\ HINTEN, & \pi - \delta_{RH} \leq \omega_{PRP} \\ SEITE, & \text{sonst} \end{cases} \quad (\text{A.3})$$

- δ_{RH} : Winkel des Aufenthaltspunktes der Person relativ zur aktuellen Bewegungsrichtung des Roboters.
- ω_{PRP} : Winkel zur Bestimmung der relativen Lage der Person zum Roboter.

Es werden die Zustände $Z_{PRP} = \{VOR, HINTEN, SEITE\}$ unterschieden. Der Parameter ω_{PRP} gilt für den gesamten Entscheidungsbaum. Die Unterscheidung der Lage ist ähnlich der Unterscheidung der Zustände bei der Bewegungsintention. Die Abbildung A.1 der Bewegungsintention kann deshalb analog dazu betrachtet werden.

A.1.3 Relative Lage zur Engstelle

Diese Bedingung dient der Bestimmung der Lage des Roboters bzw. Person relativ zur Engstelle. Es werden dabei die Zustände $Z_E = \{BEVOR, IN, NACH, \emptyset\}$ unterschieden. Die hier erläuterten Bedingungen werden von jedem Unterbaum benutzt, sodass die Parameter der Bedingungen von den Unterbäumen geteilt werden. In Abbildung A.2 werden die Bereiche schematisch dargestellt.

Die Bestimmung der relativen Lage des Roboters wird mit der Pfaddistanz a_{RS} des Roboters zum Engstellenstart bestimmt. Weil die Engstelle relativ entlang des Pfad bestimmt wird, kann der Roboter sich nicht nach einer Engstelle befinden und es gibt auch keine \emptyset -Bereiche.

Für die Personen gibt es Regionen innerhalb der Engstelle, die zum \emptyset -Bereich gehören. Dies ergibt sich aus der Bedingung $P_{In}(\delta_{HS}, \delta_{HE} | \omega_{IN})$. Sie wurde mit der Intention erstellt, Personen zu erkennen, die an den Seiten außerhalb der Engstelle sich befinden. In diesen Fall soll der Roboter die Personen als unkritisch einstufen, da es nicht möglich ist, dass eine Person sich durch die Seite bewegt, um in die Engstelle zu gelangen. Bei kurzen Engstellen und geeigneter Wahl von ω_{IN} können solche Personen erkannt werden und es führt nicht zu Fehldektionen, wenn Personen sich innerhalb der Engstelle befinden. Unter der Annahme, dass eine Engstelle seitlich nur Platz für eine Person hat, kann eine Fehldetektion für Personen innerhalb der Engstelle nur dann auftreten, wenn die Person in der Wand steht. Dies sollte physikalisch nicht möglich sein.

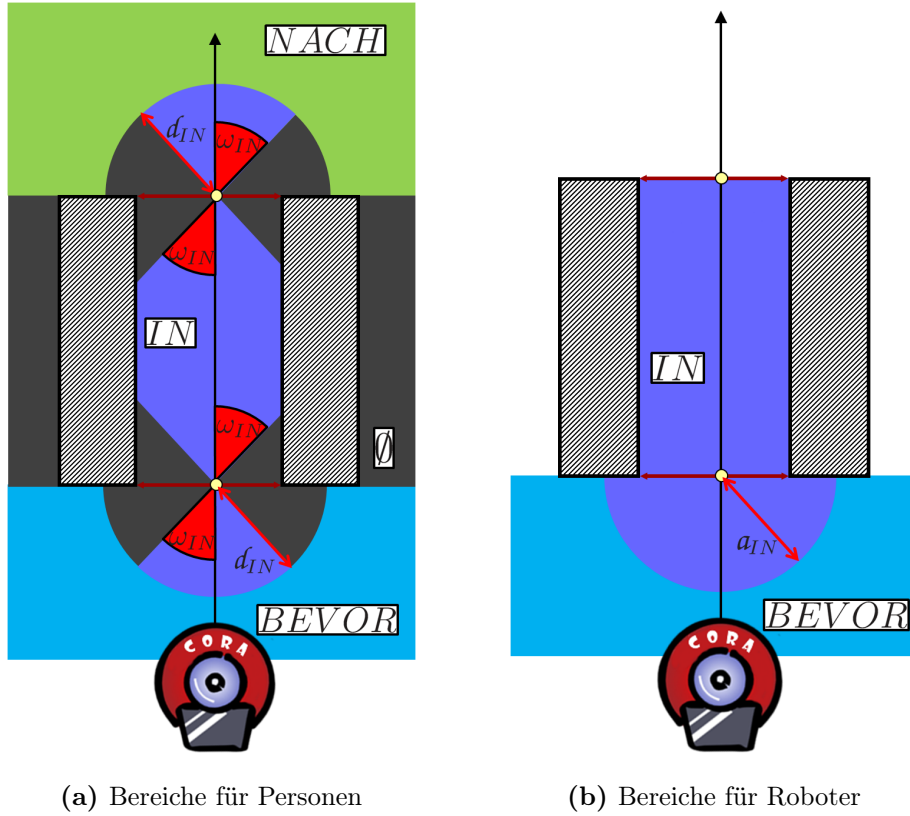


Abbildung A.2: Illustration der Bereiche bei der Bestimmung der relativen Lage vom Roboter und Personen zur Engstelle.

Aus der Definition von $P_{IN}(\delta_{HS}, \delta_{HE} | \omega_{IN})$ ergibt sich jedoch bei langen Engstellen ein Bereich an den äußeren Engstellenseiten, die als innerhalb der Engstelle erkannt werden. Aus zeitlichen Gründen konnte dies nicht weiter verfolgt werden. Es ist aber anzumerken, dass dies nur dann Einfluss auf die Erkennung hat, wenn Personen seitlich an die Engstelle sich annähern können.

In den Bereich vor und nach der Engstelle existieren \emptyset -Bereiche. Diese werden im Unterbaum für stehende Personen genutzt. Es können Situationen entstehen, in denen eine Person dort steht und mit dieser Kollisionen auftreten. In diesen Fall sollte der Roboter die Person als nicht innerhalb der Engstelle detektieren, sodass keine Reaktion ausgelöst wird.

Roboter

$$P_{Roboter, Engstelle}(a_{RS} | a_{IN}) = \begin{cases} BEVOR, & a_{RS} > a_{IN} \\ IN, & a_{RS} \leq a_{IN} \end{cases} \quad (A.4)$$

- a_{RS} : Pfaddistanz des Roboters zum Engstellenstart.
- a_{IN} : Schwellwert der Pfaddistanz zum Engstellenstart zur Bestimmung vom IN -Zustand.

Person

$$P_{Person,Engstelle}(d_{HS}, d_{HE}, \delta_{HS}, \delta_{HE} | d_{IN}, \omega_{IN}) = \begin{cases} BEVOR, & d_{HS} > d_{IN} \wedge \delta_{HS} \geq \frac{\pi}{2} \\ & P_{Pre}(d_{HS}, \delta_{HS} | d_{IN}, \omega_{IN}) \\ IN, & \vee P_{In}(\delta_{HS}, \delta_{HE} | \omega_{IN}) \\ & \vee P_{Post}(d_{HE}, \delta_{HE} | d_{IN}, \omega_{IN}) \\ NACH, & d_{HE} > d_{IN} \wedge \delta_{HE} \leq \frac{\pi}{2} \\ \emptyset, & \text{sonst} \end{cases} \quad (A.5)$$

$$P_{Pre}(d_{HS}, \delta_{HS} | d_{IN}, \omega_{IN}) = \begin{cases} 1, & d_{HS} \leq d_{IN} \wedge \pi - \delta_{HS} \leq \omega_{IN} \\ 0, & \text{sonst} \end{cases} \quad (A.6)$$

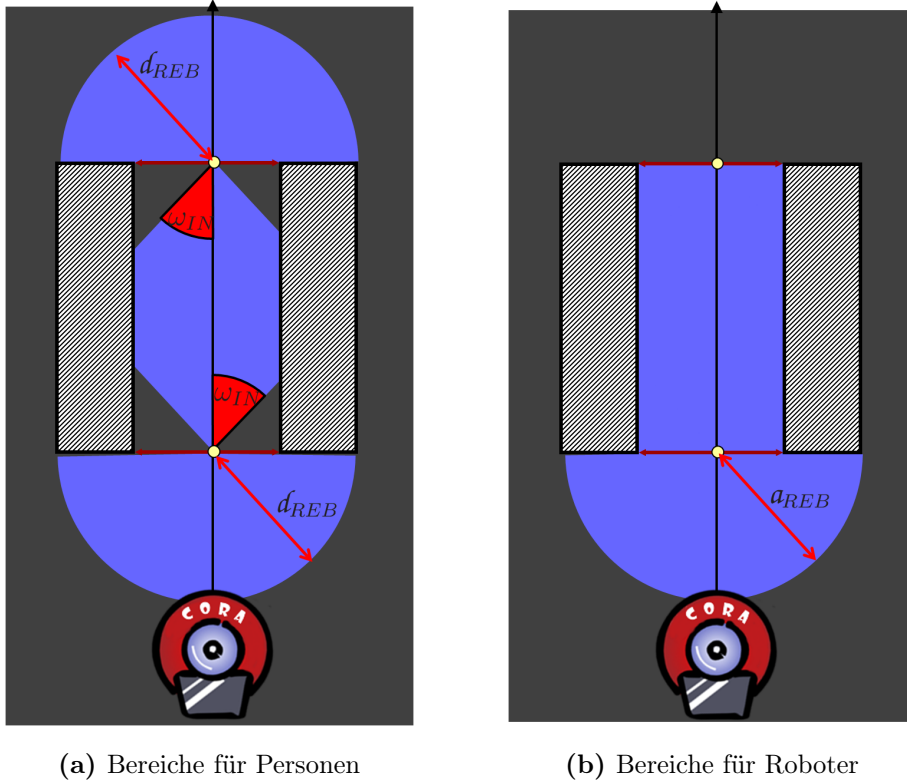
$$P_{In}(\delta_{HS}, \delta_{HE} | \omega_{IN}) = \begin{cases} 1, & \delta_{HS} \leq \omega_{IN} \wedge \pi - \delta_{HE} \leq \omega_{IN} \\ 0, & \text{sonst} \end{cases} \quad (A.7)$$

$$P_{Post}(d_{HE}, \delta_{HE} | d_{IN}, \omega_{IN}) = \begin{cases} 1, & d_{HE} \leq d_{IN} \wedge \delta_{HE} \leq \omega_{IN} \\ 0, & \text{sonst} \end{cases} \quad (A.8)$$

- d_{HS}, d_{HE} : Luftdistanz der Person zum Engstellenstart bzw. -ende.
- δ_{HS}, δ_{HE} : Winkel der Orientierung der Person relativ zur Bewegungsrichtung des Roboters am Engstellenstart bzw. -ende.
- d_{IN} : Schwellwert der Distanz zum Engstellenstart bzw. -ende zur Bestimmung vom IN -Zustand.
- ω_{IN} : Schwellwert des Winkels δ_{HS} bzw. δ_{HE} zur Bestimmung vom IN -Zustand.

A.1.4 Reaktionsbereich

Der Reaktionsbereich ist die Umgebung einer Engstelle, in denen auf Verklebungssituationen reagiert wird. Nur wenn der Roboter und die Person sich beide in diesen Reaktionsbereich befinden, soll der Roboter auf mögliche Verklebungssituationen antworten. In Abbildung A.3 ist der Reaktionsbereich dargestellt.



(a) Bereiche für Personen

(b) Bereiche für Roboter

Abbildung A.3: Illustration des Reaktionsbereiches vom Roboters und Personen. Die violettfarbenen Regionen stellen den Reaktionsbereich dar. Die grauen Regionen befindet sich außerhalb des Reaktionsbereiches.

Innerhalb der Engstelle existieren bei den Personen Regionen, die nicht zum Reaktionsbereich gehören. Für diese gilt die gleiche Argumentation wie bei der relativen Lage der Person zur Engstelle (Abschnitt A.1.3).

Roboter

$$REB_{Roboter}(a_{RS}|a_{REB}) = \begin{cases} 1, & a_{RS} \leq a_{REB} \\ 0, & a_{RS} > a_{REB} \end{cases} \quad (\text{A.9})$$

- a_{RS} : Pfaddistanz des Roboters zum Engstellenstart.
- a_{REB} : Schwellwert zur Bestimmung des Reaktionsbereiches.

Person

$$REB_{Person}(d_{HS}, d_{HE}, \delta_{HS}, \delta_{HE} | d_{REB}, \omega_{IN}) = \begin{cases} 1, & \begin{aligned} & d_{HS} \leq d_{REB} \wedge \delta_{HS} \geq \frac{\pi}{2} \\ & \vee P_{In}(\delta_{HS}, \delta_{HE} | \omega_{IN}) \\ & \vee d_{HE} \leq d_{REB} \wedge \delta_{HE} \leq \frac{\pi}{2} \end{aligned} \\ 0, & \text{sonst} \end{cases} \quad (\text{A.10})$$

- d_{HS}, d_{HE} : Luftdistanz der Person zum Engstellenstart bzw. -ende.
- δ_{HS}, δ_{HE} : Winkel der Orientierung der Person relativ zur Bewegungsrichtung des Roboters am Engstellenstart bzw. -ende.
- $P_{In}(\delta_{HS}, \delta_{HE} | \omega_{IN})$ aus Gleichung (A.7)
- d_{REB} : Schwellwert der Distanz zum Engstellenstart bzw. -ende.
- ω_{IN} : Schwellwert des Winkels δ_{HS} bzw. δ_{HE} .

A.1.5 Kritischer Kollisionsbereich

$$KKB(a_{CS}, a_{CE} | a_S, a_E) = \begin{cases} 1, & a_{CS} \geq -a_S \wedge a_{CE} \leq a_E \\ 0, & \end{cases} \quad (\text{A.11})$$

- a_{CS}, a_{CE} : Pfaddistanz der Kollision relativ zum Engstellenstart bzw. -ende. Sei a_{RC} die Pfaddistanz des Roboters zur Kollision. Dann gilt $a_{CS} = a_{RC} - a_{RS}$ und $a_{CE} = a_{RC} - a_{RE}$.
- a_S, a_E : Schwellwerte zur Bestimmung der Größe des Bereiches vor bzw. nach der Engstelle.

Diese Bedingung überprüft, ob die Kollision sich in einem kritischen Bereiche der Engstelle befindet. Dieser Bereich ist dem Reaktionsbereich ähnlich, wird jedoch mit der relativen Pfaddistanz des Roboters zur Kollision berechnet. Zudem kann die Größe der Bereiche vor und nach der Engstelle einzeln kontrolliert werden. Die Kollisionsbereiche unterscheiden sich für die einzelnen Unterbäume, sodass jeder Unterbaum ein Parametertupel (a_S, a_E) hat.

A.1.6 Vermeidung von unnötigen Reaktionen

Die folgenden Bedingung werden in den einzelnen Unterbäumen genutzt, damit unnötige Reaktionen vermieden werden. Wenn der Roboter nahe dem Engstellenende ist, sollte für ein flüssiges Vorankommen keine Reaktion ausgeführt werden. In den meisten Fällen sollten auch keine Reaktion aufgerufen werden, wenn die Kollisionen in einem unmittelbaren Bereich des Roboters und der Person liegen. In diesen Fall steht der Roboter und die Person sehr nahe beieinander. Wenn z.B. noch zusätzlich die Bedingung einer bewegenden Person erfüllt ist, dann kann davon ausgegangen werden, dass die Person auf eine Konfrontation eingestellt ist und nur an den Roboter vorbei möchte. Die Parameter dieser Bedingungen gelten für den gesamten Entscheidungsbaum.

Roboter nahe dem Engstellenende

$$NE_{Roboter}(a_{RE}|a_{NE}) = \begin{cases} 1, & a_{RE} \leq a_{NE} \\ 0, & \end{cases} \quad (\text{A.12})$$

- a_{RE} : Pfaddistanz des Roboters zum Engstellenende.
- a_{NE} : Schwellwert für Pfaddistanz des Roboters zum Engstellenende.

Roboter und Person nahe der Kollision

$$NK(a_{RC}, d_{HC}|d_{NK}) = \begin{cases} 1, & a_{RC} \leq d_{NK} \wedge d_{HC} \leq d_{NK} \\ 0, & \end{cases} \quad (\text{A.13})$$

- a_{RC} : Pfaddistanz des Roboters zur Kollision.
- d_{HC} : Luftdistanz der Person zur Kollision.
- d_{NK} : Schwellwert für die Distanzen zu den Kollisionen.

A.2 Stehende Person

In Abbildung A.4 ist der Unterbaum für stehende Personen dargestellt. Die Bedingung der PfadBlockierung kann mit dem geplanten Roboterpfad $\pi = (p_i \in \mathbb{R}^2)_{i=1,\dots,k}$ bestimmt werden. Wenn das Navigationsmodul keinen Pfad findet, wird vom Robotersystem ein leeres

Tupel ausgegeben.

A.3 Mitbewegung

In Abbildung A.5 ist der Unterbaum für Personen dargestellt, die sich in die gleiche Richtung bewegen wie der Roboter. Es werden die zusätzlichen Bedingungen MEL und NE_{Person} eingeführt. MEL überprüft die Engstelle auf eine minimale Länge und NE_{Person} überprüft die Nähe der Person zum Engstellenende. Befindet sich der Roboter vor der Engstelle und die Person innerhalb der Engstelle, so sollte der Roboter eine Einreihung nur einleiten, wenn die Engstelle eine minimale Länge hat und die Person sich nicht nahe dem Ende befindet. Dies vermeidet ein unnötiges Folgen der Person, da die Person in diesen Fall sich nicht lange in der Engstelle aufhalten wird.

Die Bedingung BD_{MIT} ist eine Konjunktion von mehreren Bedingungen. Für BD_{MIT} gilt

$$BD_{MIT} = (P_{Roboter, Person} == VOR) \wedge (P_{Person, Engstelle} == IN) \wedge \neg NE_{Roboter} \wedge \neg NE_{Person}. \quad (A.14)$$

BD_{MIT} wird überprüft, wenn der Roboter sich innerhalb der Engstelle befindet. Im Grunde besagt BD_{MIT} , dass mit einer Person nur dann eine Einreihung erfolgen soll, wenn die Person sich vor dem Roboter befindet und innerhalb der Engstelle ist. Die Bedingungen $\neg NE_{Roboter}$ und $\neg NE_{Person}$ werden nur überprüft, um eine unnötige Reaktion auszuschließen. Da der Roboter sich in der Engstelle befindet, sollte er Personen, die sich hinter ihm befinden, nicht folgen, sondern so schnell wie möglich aus der Engstelle fahren. Personen zu folgen, die sich außerhalb der Engstelle befinden und eine gleiche Bewegungsrichtung haben, ist nicht sinnvoll, da keine Verklemmungen auftreten können.

Minimale Engstellenlänge

$$MEL(a_L | \ell_{MEL}) = \begin{cases} 1, & a_L \geq \ell_{MEL} \\ 0, & \end{cases} \quad (A.15)$$

- a_L : Länge der Engstelle.
- ℓ_{MEL} : Schwellwert für die minimale Länge einer Engstelle.

Person nahe dem Engstellenende

$$NE_{Person}(d_{HE}|d_{NE}) = \begin{cases} 1, & d_{HE} \leq d_{NE} \\ 0, & \end{cases} \quad (\text{A.16})$$

- d_{HE} : Luftdistanz der Person zum Engstellenende.
- d_{NE} : Schwellwert für Distanz der Person zum Engstellenende.

A.4 Gegenbewegung

In Abbildung A.6 ist der Unterbaum für Personen, die den Roboter entgegenkommen. Für diesen Unterbaum werden die zusätzlichen Bedingungen NS_{Person} und VFR benötigt. NS_{Person} überprüft, ob die Person sich nahe dem Engstellenstart befindet. VFR realisiert eine Vorfahrtregel bezogen auf das Engstellenende.

Die Bedingung BD_{GEGEN} ist eine Konjunktion von mehreren Bedingungen. Für BD_{GEGEN} gilt

$$BD_{GEGEN} = KKB_{GEGEN} \wedge \neg NK \wedge \neg NE_{Roboter} . \quad (\text{A.17})$$

Mit BD_{GEGEN} wird überprüft, ob sich die Kollision im kritischen Bereich befindet und es werden mit NK und $NE_{Roboter}$ unnötige Reaktionen erkannt.

Die Bedingung NS_{Person} wird überprüft, wenn der Roboter sich vor der Engstelle befindet und die Person sich innerhalb der Engstelle. Damit wird ein unnötiges Warten des Roboters vermieden, wenn die Person sich nahe dem Engstellenstart befindet. Da die Person in Bewegung ist, wird sie die Engstelle bald verlassen und den Weg für den Roboter freigeben.

Die Vorfahrtregel wird überprüft, wenn der Roboter sich innerhalb der Engstelle befindet und die Person nach der Engstelle. Zudem befindet sich der Roboter wegen BD_{GEGEN} nicht nahe dem Engstellenende. In diesen Fall wird angenommen, dass die Engstelle relativ lang ist. Mit VFR kann dann bestimmt werden, ob der Roboter durch die Engstelle fährt und die Person warten lässt oder ob er der Person die Vorfahrt gewährt.

Vorfahrtregel

$$VFR_{Roboter}(a_{RE}, d_{HE} | d_{VFR}) = \begin{cases} 1, & d_{HE} > a_{RE} + d_{VFR} \\ 0, & \text{sonst} \end{cases} \quad (\text{A.18})$$

- a_{RE} : Pfaddistanz des Roboters zum Engstellenende.
- d_{HE} : Luftdistanz der Person zum Engstellenende.
- d_{VFR} : Bias zur Steuerung der Vorfahrtregel. Bei großen positiven Werten neigt der Roboter dazu die Vorfahrt zu gewähren, obwohl er näher am Engstellenende ist. Bei negativen Werten neigt der Roboter dazu seine Vorfahrt einzufordern, obwohl die Person näher am Engstellenende ist.

Person nahe dem Engstellenstart

$$NS_{Person}(d_{HS} | d_{NS}) = \begin{cases} 1, & d_{HS} \leq d_{NS} \\ 0, & \text{sonst} \end{cases} \quad (\text{A.19})$$

- d_{HS} : Luftdistanz der Person zum Engstellenstart.
- d_{NS} : Schwellwert für Distanz der Person zum Engstellenstart.

A.5 Tabelle der Parameter

Parameter	Standardwert	Beschreibung
<i>BI</i> : Bewegungsintention		
v_{min}	$0.2 \frac{m}{s}$	Minimale Geschwindigkeit einer sich bewegenden Person.
ω_{BI}	$1.39 (80^\circ)$	Winkel zur Bestimmung der Bewegungsrichtung der Person.
$P_{Roboter, Person}$: Relative Lage der Person zum Roboter		
ω_{PRP}	$1.39 (80^\circ)$	Winkel zur Bestimmung der relativen Lage der Person zum Roboter.
$P_{Roboter, Engstelle}$: Relative Lage des Roboters zur Engstelle		
a_{IN}	1 m	Distanz zum Engstellenstart zur Bestimmung des Aufenthalts des Roboters innerhalb der Engstelle.

$P_{Person,Engstelle}$: Relative Lage der Person zur Engstelle		
d_{IN}	1 m	Distanz zum Engstellenstart bzw. -ende zur Bestimmung des Aufenthalts der Person innerhalb der Engstelle.
ω_{IN}	1.22 (70°)	Winkels zur Bestimmung der Orientierung zum Engstellenstart bzw. -ende für die Bestimmung des Aufenthalts der Person innerhalb der Engstelle.
$REB_{Roboter}$: Reaktionsbereich des Roboters		
a_{REB}	4 m	Parameter zur Bestimmung der Größe des Reaktionsbereiches.
REB_{Person} : Reaktionsbereich der Person		
d_{REB}	4 m	Parameter zur Bestimmung der Größe des Reaktionsbereiches.
ω_{IN}	-	Identisch mit ω_{IN} von $P_{Person,Engstelle}$.
$KKB_{STEHEND}$: Kritischer Kollisionsbereich bei stehenden Personen		
$a_{S,STEHEND}$	1 m	Distanz zur Bestimmung der Größe des kritischen Bereiches vor der Engstelle.
$a_{E,STEHEND}$	1 m	Distanz zur Bestimmung der Größe des kritischen Bereiches nach der Engstelle.
KKB_{MIT} : Kritischer Kollisionsbereich bei mitlaufenden Personen		
$a_{S,MIT}$	2 m	Distanz zur Bestimmung der Größe des kritischen Bereiches vor der Engstelle.
$a_{E,MIT}$	4 m	Distanz zur Bestimmung der Größe des kritischen Bereiches nach der Engstelle.
KKB_{GEGEN} : Kritischer Kollisionsbereich bei entgegenkommenden Personen		
$a_{S,GEGEN}$	2 m	Distanz zur Bestimmung der Größe des kritischen Bereiches vor der Engstelle.
$a_{E,GEGEN}$	2 m	Distanz zur Bestimmung der Größe des kritischen Bereiches nach der Engstelle.
$NE_{Roboter}$: Roboter nahe dem Engstellenende		
a_{NE}	1 m	Distanz zur Bestimmung der Nähe zum Engstellenende.
NE_{Person} : Person nahe dem Engstellenende (Nutzung nur im Unterbaum der Mitbewegung)		
d_{NE}	0.5 m	Distanz zur Bestimmung der Nähe zum Engstellenende.
NS_{Person} : Person nahe dem Engstellenstart (Nutzung nur im Unterbaum der Gegenbewegung)		

d_{NS}	0.5 m	Distanz zur Bestimmung der Nähe zum Engstellenstart.
<i>NK</i> : Roboter und Person nahe der Kollision		
d_{NK}	0.4 m	Distanzen zur Bestimmung der Nähe zu den Kollisionen.
<i>MEL</i> : Minimale Engstellenlänge (Nutzung nur im Unterbaum der Mitbewegung)		
l_{MEL}	1 m	Schwellwert für die minimale Länge einer Engstelle.
<i>VFR</i> : Vorfahrtsregel (Nutzung nur im Unterbaum der Gegenbewegung)		
d_{VFR}	0.2 m	Bias zur Steuerung der Vorfahrtregel. Bei großen positiven Werten neigt der Roboter dazu die Vorfahrt zu gewähren, obwohl er näher am Engstellenende ist. Bei negativen Werten neigt der Roboter dazu seine Vorfahrt einzufordern, obwohl die Person näher am Engstellenende ist.

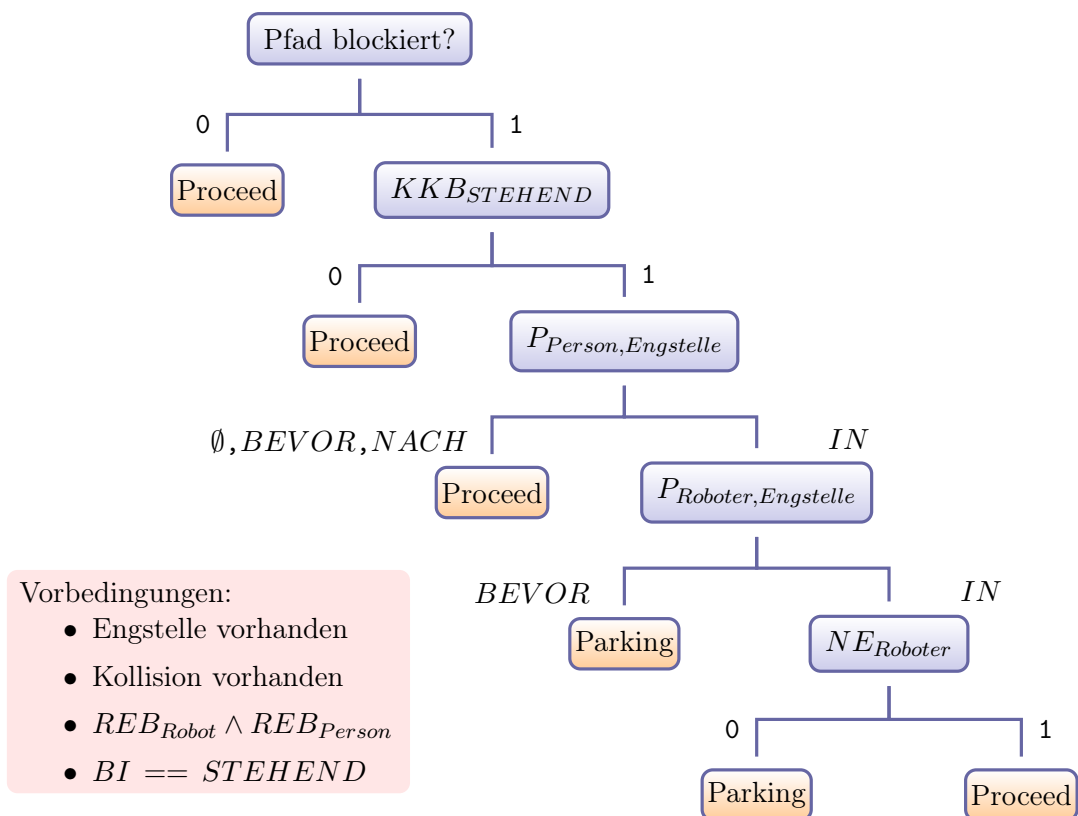


Abbildung A.4: Unterbaum für stehende Personen. Für eine nähere Erläuterung wird auf Abschnitt A.2

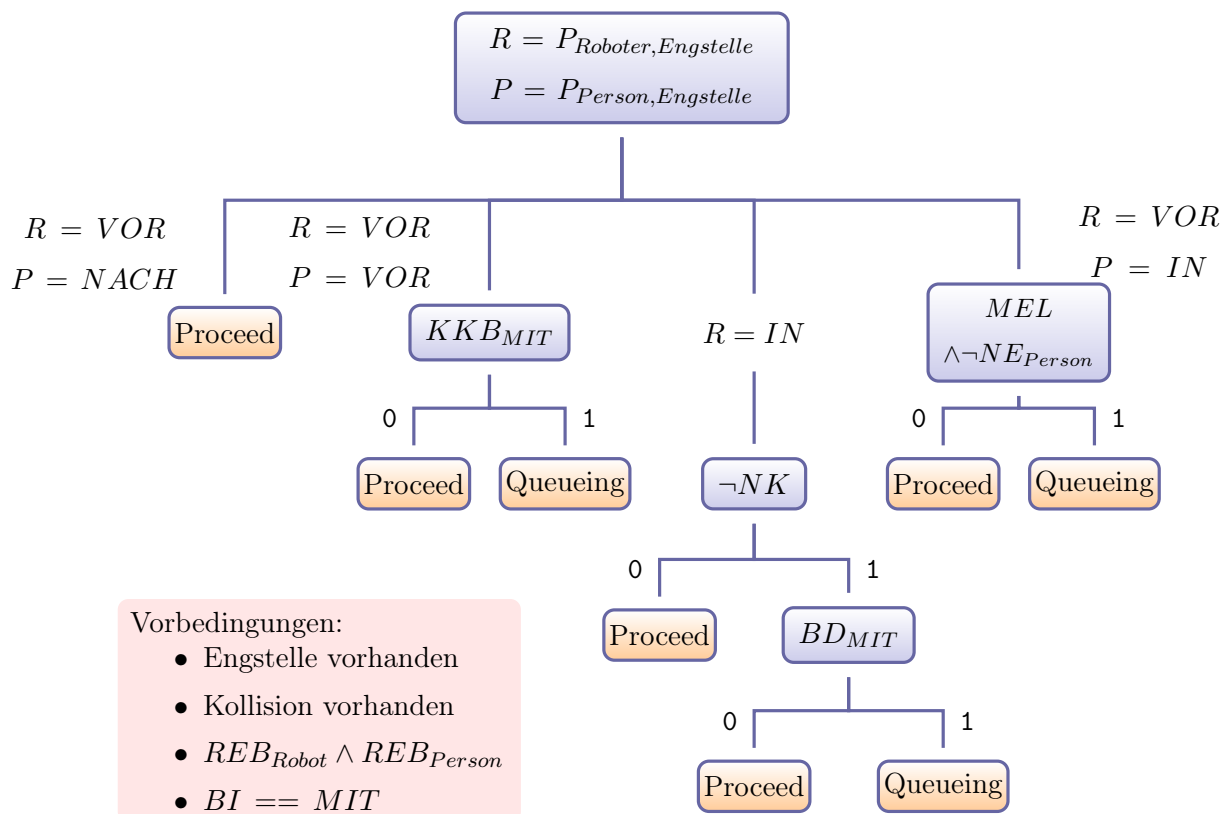


Abbildung A.5: Unterbaum für mitlaufende Personen. Für eine nähere Erläuterung wird auf Abschnitt A.3.

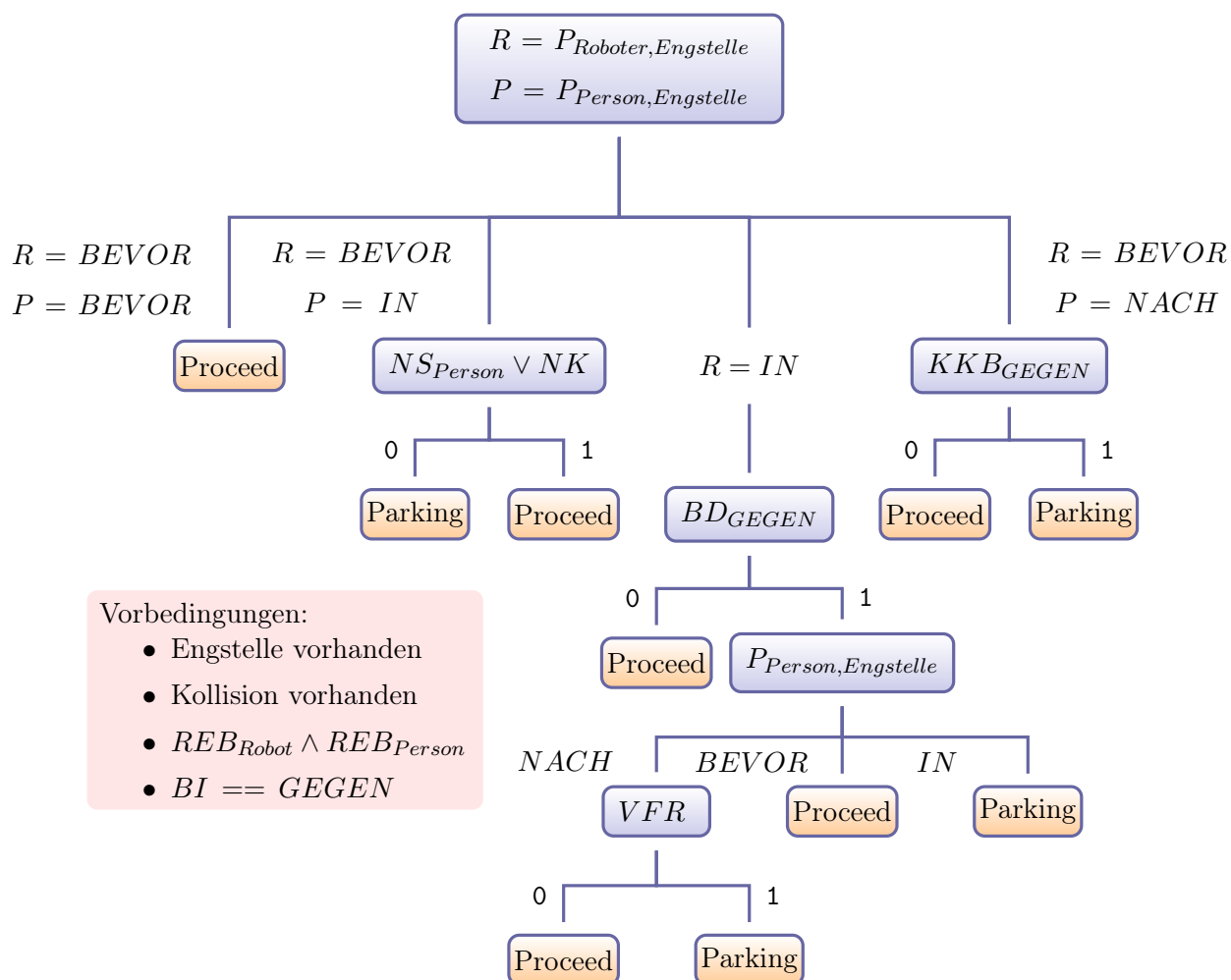


Abbildung A.6: Unterbaum für entgegenkommende Personen. Für eine nähere Erläuterung wird auf Abschnitt A.4.

Anhang B

Ergänzende Unterlagen

B.1 Konfusionsmatrix

		Ground Truth (GT)			
		L_1	L_2	\dots	L_n
Prädizierte Klasse (PK)	L_1	c_{11}	c_{12}		c_{1n}
	L_2	c_{21}	c_{22}	\dots	c_{2n}
	\dots		\dots		\dots
	L_n	c_{n1}	c_{n2}	\dots	c_{nn}

Abbildung B.1: Eine Konfusionsmatrix $K \in \mathbb{N}^{(n,n)}$ mit n Klassen. L_i bezeichnet dabei die Klassenlabels. c_{ij} ist die Anzahl der Merkmalsvektoren von der Klasse L_j , die vom Klassifikator der Klasse L_i zugeordnet wurde.

Für die Bewertung der Performance wird eine *Konfusionsmatrix* für mehrere Klassen verwendet. Sie enthält die vom Klassifikator getroffenen Entscheidungen in einer geordneten Darstellung. In Abbildung B.1 ist eine Konfusionsmatrix für n Klassen dargestellt. Sei $C : \mathbb{R}^n \rightarrow \{1, \dots, n\}$ ein Klassifikator, der n Klassen unterscheiden kann und sei $G = \{(\mathbf{x}_i, l_i) | \mathbf{x}_i \in \mathbb{R}^n, l_i \in \{1, \dots, n\}, 1 \leq i \leq N\}$ ein gelabelter Testdatensatz. Über den Datensatz G lässt sich die Konfusionsmatrix des Klassifikators C mit folgender Eigenschaft der Matrixelemente berechnen:

$$c_{ij} = |\{(\mathbf{x}, l) \in G | l = i \wedge C(\mathbf{x}) = j\}| \quad (\text{B.1})$$

Die Summe einer Matrixreihe i entspricht folglich der Anzahl der Instanzen, die im Datensatz

zu der Klasse i gehören. Die Summe der Werte einer Spalte j entspricht dagegen der Anzahl der Instanzen, die vom Klassifikator der Klasse j zugeordnet wurden.

Ausgehend von der Konfusionsmatrix wird zudem für jede Klasse i die *Richtig-Positiv-Rate* TPR_i mit

$$TPR_i = \frac{c_{ii}}{\sum_{k=1}^n c_{ki}} \quad (\text{B.2})$$

berechnet werden. Sie entspricht dem Anteil der richtig klassifizierten Instanzen einer Klasse.

B.2 Quellenangabe

KONFERENZEN						
Jahr	09	10	11	12	13	14
SMC - IEEE International Conference on Systems, Man, and Cybernetic	0	0	0	0	0	1
IROS - International Conference Intelligent Robots and Systems	0	0	0	1	0	0
FUSION - International Conference on Information Fusion	0	0	1	0	0	0
ROMAN - International Symposium on Robot and Human Interactive Communication	1	0	0	0	0	0
ICTAI - International Conference on Tools with Artificial Intelligence	0	1	0	0	0	0
AIM - International Conference on Advanced Intelligent Mechatronics	0	0	1	0	0	0
ITS - World Congress on Intelligent Transportation Systems	1	0	0	0	0	0

Abbildungsverzeichnis

1.1	Luftaufnahme der Fachklinik Bad Liebenstein	2
1.2	Grundriss der Fachklinik Bad Liebenstein mit	3
2.1	Schematische Darstellung des entwickelten Verfahren zur Situationserkennung	5
2.2	Klassen von Verklemmungssituationen	6
2.3	Schichtenarchitektur des Softwaresystems von ROREAS	9
2.4	Zusammenwirken des Situation Classifier mit der Control Layer	10
2.5	Blockdiagramm zum Aufbau des Situation Classifiers	11
3.1	Schematische Darstellung der Prädiktion von Platzkonflikten	16
3.2	Merkmale zur Beschreibung von Platzkonflikten mit Personen	17
3.3	Schematische Darstellung der Engstellenerkennung	22
3.4	Merkmale zur Beschreibung der Engstelle in Zusammenhang mit dem Roboter und einer kollidierenden Person	23
3.5	Probleme vom linearen Spline	26
3.6	Kubische Smoothing Splines mit unterschiedlichen Glättungsparameter	27
3.7	Smoothing Splines bei kurvigen Pfaden und Schwankungen in den Pfadpunkten	28
4.1	Grundprinzip der SVM zur Trennung von zwei Klassen	30
4.2	Darstellung des Soft-Margin und die Support Vektoren einer Trenngerade . . .	31
4.3	Top-Level Ebenen des entwickelten Entscheidungsbaum	33
5.1	Visualisierungen der verwendeten Merkmale zum Labeln der Aufnahmen	36
5.2	Grundriss der Aufnahmeumgebung	37
5.3	Darstellung einer nichtlinearen Trenngerade eines Entscheidungsbaumes	42

6.1	Beispiel für ein fälschliche Verklemmungssituation durch Vernachlässigung der Umgebungskarte.	49
6.2	Beispiel für eine Maskierung von konfliktären Kollisionen.	50
6.3	Mehrdeutige Situation durch ungenaue Schätzung der Bewegungsintention . .	51
6.4	Beispiel einer fälschlichen Verklemmungssituation unter Nichtbeachtung der Personenbewegung.	52
A.1	Winkelbereiche zur Bestimmung der Bewegungsintention einer Person	58
A.2	Darstellung der Bereiche bei der Bestimmung der relativen Lage vom Roboter und Personen zur Engstelle	60
A.3	Darstellung der Bereiche bei der Bestimmung des Reaktionsbereiches vom Roboter und Personen	62
A.4	Unterbaum für stehende Personen	70
A.5	Unterbaum für mitlaufende Personen	71
A.6	Unterbaum für entgegenkommende Personen	72
B.1	Konfusionsmatrix für n Klassen	73

Literaturverzeichnis

- [ANATOLYEVICH, 2014] ANATOLYEVICH, B.S. (2014). *ALGLIB - Numerical Analysis Library*. <http://www.alglib.net> (aufgerufen am 14.09.2014).
- [AYARI et al., 2010] AYARI, E., S. HADOUAJ und K. GHEDIRA (2010). *A Multi-agent Simulation Model Based on Fuzzy Logic to Predict Conflict Situations for Autonomous Robot Navigation*. In: *22nd IEEE International Conference on Tools with Artificial Intelligence (ICTAI)*, Bd. 1, S. 193–196.
- [BECK et al., 2011] BECK, AB., C. RISAGER, N. ANDERSEN und O. RAVN (2011). *Spacio-temporal situation assessment for mobile robots*. In: *Proceedings of the 14th International Conference on Information Fusion (FUSION)*, S. 1–8.
- [CHANG und LIN, 2011] CHANG, CHIH-CHUNG und C.-J. LIN (2011). *LIBSVM: A library for support vector machines*. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27. Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.
- [DAKULOVIC et al., 2011] DAKULOVIC, M., B. DIMITRIJEVIC und I. PETROVIC (2011). *Avoiding deadlocks of mobile robots in narrow passages of environments populated with moving obstacles*. In: *IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, S. 936–941.
- [DE BOOR, 2001] DE BOOR, CARL (2001). *A practical guide to splines: with 32 figures*. Nr. v. 27 in *Applied mathematical sciences*. Springer, New York, Rev. ed Aufl.
- [EINHORN et al., 2012] EINHORN, E., T. LANGNER, R. STRICKER, C. MARTIN und H. GROSS (2012). *MIRA - middleware for robotic applications*. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, S. 2591–2598.
-

- [GROSS et al., 2014] GROSS, H.-M., K. DEBES, E. EINHORN, S. MUELLER, A. SCHEIDIG, C. WEINRICH, A. BLEY und C. MARTIN (2014). *Mobile Robotic Rehabilitation Assistant for walking and orientation training of Stroke Patients: A report on work in progress*. In: *IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, S. 1880–1887.
- [KÜHNEL, 2008] KÜHNEL, WOLFGANG (2008). *Differentialgeometrie: Kurven - Flächen - Mannigfaltigkeiten*. Vieweg, Wiesbaden.
- [MATLAB, 2014] MATLAB (2014). *Curve Fitting Toolbox Dokumentation - Version 8.3.0 (R2014a)*. <http://www.mathworks.de> (aufgerufen am 14.09.2014).
- [RENDON-VELEZ et al., 2009] RENDON-VELEZ, E., I. HORVÁTH und E. Z. OPIYO (2009). *Progress with situation assessment and risk prediction in advanced driver assistance systems: A survey*. In: *Proceedings of the 16th ITS World Congress*.
- [STATNIKOV et al., 2011] STATNIKOV, A., C. ALIFERIS und D. HARDIN (2011). *A Gentle Introduction to Support Vector Machines in Biomedicine: Theory and methods*. A Gentle Introduction to Support Vector Machines in Biomedicine. World Scientific.
- [TRANBERG HANSEN et al., 2009] TRANBERG HANSEN, S., M. SVENSTRUP, H. ANDERSEN und T. BAK (2009). *Adaptive human aware navigation based on motion pattern analysis*. In: *The 18th IEEE International Symposium on Robot and Human Interactive Communication (ROMAN)*, S. 927–932.
-

Thesen

- In einer dynamischen Klinikumgebung können Verklemmungssituationen entstehen, die durch Engstellen ausgelöst werden.
- Für eine höfliche und nutzerzentrierte Roboternavigation müssen Verklemmungssituationen im Voraus erkannt werden, um rechtzeitig eine Reaktion einzuleiten.
- Engstellensituationen können durch reellwertige Merkmale beschrieben werden, die aus dem geplanten Robterpfad und dem Bewegungszustand der umgebenden Personen extrahiert werden.
- Durch Nutzung eines Klassifikators können Verklemmungssituationen in den Merkmalsraum von unkritischen Situationen unterschieden werden.
- Ein per Hand designer Entscheidungsbaum und eine lineare Support Vektor Maschine können zur Klassifikation in den entwickelten Merkmalsraum eingesetzt werden.
- Es wurde experimentell gezeigt, dass die entwickelten Merkmale zur Erkennung von Verklemmungssituationen geeignet sind.
- Auf dem verwendeten Testdatensatz erzielte der Entscheidungsbaum eine deutlich bessere Erkennungsrate als die lineare Support Vektor Maschine.

Ilmenau, 23.09.2014

.....

Thanh Quang Trinh