

Fakultät für Elektrotechnik und Informationstechnik

Lehrstuhl für Regelungssystemtechnik

**Diplomarbeit**

**Überwachtes Lernen bildbasierter Roboterverhalten**

von  
**Arne Nordmann**

Verantwortlicher Hochschullehrer: Prof. Dr.-Ing. Prof. h. c. Torsten Bertram

Wissenschaftliche Betreuer: Dr. rer. nat. Frank Hoffmann  
M. Sc. Krishna Kumar Narayanan

Dortmund, 10. Juli 2009



## Kurzfassung

Die vorliegende Arbeit entwickelt ein bildbasiertes Roboterverhalten zur Korridorfolge und Kollisionsvermeidung in Innenraum-Umgebungen mit Hilfe von Methoden des maschinellen Lernens. Die Umfeldwahrnehmung basiert auf einem System aus einer Farbkamera, einem Time-of-Flight-Sensor und einer omnidirektionalen Kamera. Die Bilder in der Frontansicht und in der Rundumansicht werden durch Fusion der Bild- und Abstandsinformation segmentiert und in die drei Kategorien „Flur“, „Wand“ und „Hindernis“ unterschieden.

Die Bildinformation wird auf für das Erlernen des Roboterverhaltens relevante Merkmale wie Textur, Farbe und Form der segmentierten Teilregionen reduziert. Mit Hilfe eines instanzbasierten Lernverfahrens wird dem Roboter aus den Trainingsdaten ein allgemeines Verhalten angelehrt. Die Robustheit des erlernten Roboterverhaltens gegenüber Veränderungen der Umgebung wird anhand experimenteller Ergebnisse in unterschiedlichen realen Umgebungen untersucht.



# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
<b>2</b>	<b>Grundlagen</b>	<b>5</b>
2.1	Mobiler Roboter . . . . .	5
2.1.1	MobileRobots Pioneer . . . . .	6
2.1.2	ToF-Sensor / PMD . . . . .	6
2.1.3	Omnidirektionale Kamera . . . . .	8
2.2	Kamerakalibrierung . . . . .	12
2.3	Überwachtes maschinelles Lernen . . . . .	13
2.3.1	Künstliche neuronale Netze . . . . .	13
2.3.2	Instanzbasiertes Lernen . . . . .	14
2.3.3	Vergleich der Lernverfahren . . . . .	17
<b>3</b>	<b>Wahrnehmung</b>	<b>19</b>
3.1	Segmentierung im Raum – 3D . . . . .	21
3.1.1	Transformation . . . . .	22
3.1.2	Finden von Ebenen mittels RANSAC . . . . .	22
3.1.3	Klassifizierung der Ebenen . . . . .	25
3.2	Segmentierung im Bild – 2D . . . . .	25
3.2.1	Projektion der PMD-Messung . . . . .	26
3.2.2	Segmentierungsverfahren . . . . .	27
3.2.3	Labeln der Bilder . . . . .	30
3.3	Automatisches Bewerten der Segmentierung . . . . .	30
3.4	Omnidirektionale Ansicht . . . . .	32
3.4.1	Transformation . . . . .	33
3.4.2	Segmentierung . . . . .	34
<b>4</b>	<b>Roboterverhalten</b>	<b>35</b>
4.1	Ausgangsgröße . . . . .	36
4.2	Auswahl geometrischer Merkmale . . . . .	37
4.2.1	Manuelle Auswahl . . . . .	37
4.2.2	Automatische Auswahl . . . . .	40
4.3	Aufzeichnung von Trainingsdaten . . . . .	42
4.3.1	Manuelle Fahrt . . . . .	43
4.3.2	Autonome Fahrt . . . . .	43
4.4	Modellbildung . . . . .	44

4.4.1	Regression . . . . .	45
4.4.2	Clustering . . . . .	45
4.5	Validierung . . . . .	46
<b>5</b>	<b>Experimentelle Ergebnisse</b>	<b>49</b>
5.1	Robustheit der Wahrnehmung . . . . .	49
5.1.1	Falsch Positive . . . . .	51
5.1.2	Falsch Negative . . . . .	52
5.1.3	Fazit . . . . .	53
5.2	Roboterverhalten . . . . .	54
5.2.1	Merkmalsauswahl . . . . .	56
5.2.2	Lernmodell . . . . .	58
5.2.3	Clustering . . . . .	60
5.3	Gesamtsystem . . . . .	61
<b>6</b>	<b>Zusammenfassung und Ausblick</b>	<b>65</b>
<b>A</b>	<b>Anhang</b>	<b>77</b>
<b>B</b>	<b>Sonstiges</b>	<b>79</b>
B.1	Eigenständigkeitserklärung . . . . .	79
B.2	Aufgabenstellung . . . . .	80

# Kapitel 1

## Einleitung

### Ausgangssituation

Für Serviceroboter, die sich in Innenraum-Umgebungen bewegen und mit Menschen interagieren, ist die visuelle Wahrnehmung der Umgebung von Bedeutung und damit einhergehend die Rekonstruktion oder Repräsentation der dreidimensionalen Geometrie der Umgebung. Diese dient zum Beispiel der Lokalisation des Roboters, der Objekterkennung und der Kollisionsvermeidung. Der Roboter wird dadurch zum einen in die Lage versetzt, sich kollisionsfrei in seiner Umgebung zu bewegen und zum anderen mit Objekten in seiner Umgebung zu interagieren.

Zur Erkennung der dreidimensionalen Umwelt steht modernen Servicerobotern eine Vielzahl an Sensoren zur Verfügung. Dazu zählen Sonar-Sensoren, Radar-Sensoren, Laserscanner und Stereokamerasysteme. Viele dieser Systeme haben jedoch deutliche Nachteile. Laserscanner tasten über einen Laserstrahl, der von einem beweglichen Spiegel abgelenkt wird, zeilenweise die Umgebung ab. Der bewegliche Spiegel macht sie gegenüber Systemen ohne bewegliche Teile aufwendiger und störanfälliger. Laserscanner sind darüber hinaus teuer und bieten erst in der Preisklasse jenseits der fünfstelligen Euro-Beträge 3D-Informationen in mehr als nur einer Schnittebene. Stereokamerasysteme sind günstig in der Anschaffung, sind aber auf eine aufwendige korrespondenzbasierten Tiefenschätzung angewiesen. Für die Korrespondenzsuche sind sie zudem zwingend auf Textur in der Szene angewiesen. Sonar- und Radarsensoren bieten nur eine geringe räumliche Auflösung und abseits der Tiefeninformation keine weiteren nützlichen Informationen. Sie sind damit in der Anwendung deutlich beschränkt.

Die vorliegende Arbeit untersucht und entwickelt ein Verfahren, mit dem grundlegende Aufgaben der Servicerobotik ausschließlich basierend auf bildgebender Sensorik unter Vermeidung der eben genannten Sensortypen möglich ist. Der Vorteil eines alleinig auf bildgebender Sensorik basierenden Systems ist, dass die Anzahl der Sensoren und vor allem unterschiedlicher Sensoren reduziert wird, da man diejenigen Sensoren nutzt, die das größte Spektrum möglicher Anwendungen aufweisen. Darüber hinaus nutzt man eine Technik, die der Wahrnehmung des Menschen, in dessen Umgebung Serviceroboter operieren und dessen wichtigste Wahrnehmung die visuelle ist, sehr ähnlich ist.

Ein bisheriges Problem rein bildbasierter Wahrnehmung ist die Robustheit. Robustheit ist die

Fähigkeit, die Funktionsfähigkeit eines Systems auch bei Variation der Umgebungsbedingungen aufrecht zu erhalten. In Innenraum-Szenarien sind dies Variationen in der Beleuchtung sowie deutliche Änderungen hinsichtlich Farbe, Geometrie und Textur der Umgebung. Ansätze zur rein bildbasierten Servicerobotik in Innenraum-Szenarien beruhen bislang hauptsächlich auf Stereokamerasystemen mit den genannten Problemen und Nachteilen.

Ein relativ neuer bildgebender Sensortyp ist die 3D-Kamera, die auf der Laufzeitmessung eines Lichtsignals beruht. 3D-Kameras ermöglichen eine direkte und berührungslose Messung von Tiefeninformationen eines Bildausschnittes. Einer ihrer Vorteile ist, dass sie wie normale Kameras auf der CMOS-Technik beruhen und somit erwartet wird, dass sich ihre Kosten langfristig denen normaler Kameras annähern. Sie bieten außerdem Tiefeninformationen nicht nur in einer Schnittebene, sondern – ähnlich normalen Kameras – in einem zweidimensionalen Bildbereich. Neben den Tiefeninformationen wird darüber hinaus ein Intensitätsbild aufgenommen, das die Zuordnung der Tiefeninformationen zur optischen Wahrnehmung ermöglicht.

## Zielsetzung

Zielsetzung dieser Arbeit ist die Entwicklung eines Verfahrens zur robusten Navigation in Innenraum-Szenarien. Grundlegendes Instrument dafür ist die Kollisionsvermeidung, die dem Roboter ermöglicht, seine Bewegungen in der Umwelt ohne Kollision und damit ohne Beschädigung des Roboters oder der Umwelt durchzuführen. Darauf aufbauend ist für Innenraum-Szenarien die Fähigkeit, Korridoren zu folgen, eine essentielle, um dem Roboter zielgerichtete Fortbewegung und effiziente Wegführung zu ermöglichen.

Ziel der Arbeit ist es dementsprechend als Grundlage für die Navigation in Innenraum-Szenarien ein Roboterverhalten zu erzeugen, das diese beiden Funktionen zeigt. Die Verhalten werden dabei nicht getrennt betrachtet, sondern in ein allgemeines Verhalten zur robusten Kollisionsvermeidung und Korridorfolge vereint.

Roboterverhalten basierend auf bildgebender Sensorik sind insofern eine Herausforderung, dass die Wahrnehmung mit Kameras hoch-dimensional ist. Ein darauf basierendes Roboterverhalten in Form einfacher Stimulus-Reaktion-Regeln zu implementieren, setzt eine Reduktion der Wahrnehmung auf wenige Dimensionen voraus, mit dem der Verlust wahrgenommener Informationen einhergeht.

Eine Alternative dazu stellt das maschinelle Lernen des Roboterhaltens basierend auf der hoch-dimensionalen Wahrnehmung dar. Dabei müssen keine Regeln explizit vorgegeben werden, sondern diese werden mit Methoden des maschinellen Lernens aus den Trainingsdaten erzeugt. Der Roboter bildet dadurch selbstständig Bewegungsmuster aufgrund unterschiedlicher Ausprägungen der Umwelt aus. Explizites Vorgeben von Regeln für den Roboter durch Experten und explizites Domänenwissen ist dadurch nicht erforderlich.



## Vorgehensweise

Um Kollisionsvermeidung und Korridorfolge basierend auf bildgebender Sensorik zu ermöglichen, ist als Grundlage die robuste Wahrnehmung und Repräsentation der Umgebung erforderlich, die dem Roboter sowohl das Erkennen von Hindernissen als auch das Erkennen eines Korridors ermöglicht. Dazu wird ein Experimentalsystem aus einer in Fahrtrichtung des Roboters gerichteten monokularen Kamera und einer 3D-Kamera aufgebaut. Darüber hinaus besitzt der Roboter eine omnidirektionale Kamera, die eine 360°-Rundumsicht der Umgebung des Roboters bietet.

In der Frontansicht des Roboters werden die Ansichten der 3D-Kamera und der monokularen Kamera fusioniert, um eine genaue und robuste Segmentierung und Klassifizierung der lokalen Umgebung in Boden, Wand und freistehende Hindernisse zu erhalten. Die segmentierte und klassifizierte Frontansicht wird anhand ihrer Ähnlichkeit in die omnidirektionale Ansicht und damit auf die gesamte Umgebung des Roboters übertragen.

Reduziert auf eine geeignete Darstellung der Umgebung dient dies als Grundlage für das Erlernen des Roboterhaltens. Dazu werden mit dem Roboter Trainingsfahrten aufgezeichnet, aus deren Trainingsbeispielen mit Hilfe maschinellen Lernens das allgemeine Roboterverhalten abstrahiert wird. Zum Ende der Arbeit wird die Fähigkeit des Systems, robust gegenüber Veränderungen der Umgebung das trainierte Verhalten nachzubilden, in unterschiedlichen realen Umgebungen analysiert und verifiziert.



# Kapitel 2

## Grundlagen

Im Folgenden werden technische Grundlagen zum Verständnis der vorliegenden Arbeit erläutert.

### 2.1 Mobiler Roboter

Die vorliegende Arbeit setzt die Ergebnisse auf einem mobilen Roboter um. Grundlage ist eine mobile Plattform des Herstellers *MobileRobots Inc*, die um bildgebende Sensorik erweitert ist. Die Sensorik besteht aus einer 3D-Kamera, einer monokularen Kamera und einer omnidirektionalen Kamera. Abbildung 2.1 zeigt den mobilen Roboter in der Frontansicht.

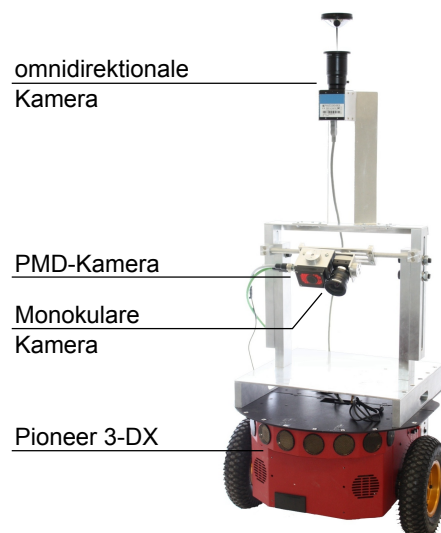


Abbildung 2.1: Das Experimentalsystem mit 3D-Kamera, monokularer Kamera und omnidirektionaler Kamera.

### 2.1.1 MobileRobots Pioneer

Die Plattform des mobilen Roboters bildet ein *Pioneer 3-DX* der Firma *MobileRobots*. Der Roboter verfügt über zwei Antriebsräder, integrierte Akkus für den Betrieb sowie 16 Sonar-Sensoren für die Umfeldwahrnehmung. Der Roboter wird außerdem über ein Castor-Rad gestützt und verfügt über Odometrie-Sensoren in den zwei Antriebsrädern zur Aufzeichnung der gefahrenen Strecke. Zur Ansteuerung via PC besitzt der *Pioneer 3-DX* einen *EIA-232*-Anschluss, über den die Sensoren ausgelesen und Motorkommandos an den Roboter gesendet werden können.

Software-seitig steht mit *ARIA (Advanced Robotics Interface for Applications)* eine Open-Source-Schnittstelle für C++ vom Hersteller zur Verfügung, über die mit dem Roboter kommuniziert werden kann. Am Lehrstuhl für Regelungssystemtechnik der TU Dortmund ist darüber hinaus eine *ARIA*-Schnittstelle in MatLab verfügbar.

### 2.1.2 ToF-Sensor / PMD

Die verwendete 3D-Kamera ist ein *Time-of-Flight-Sensor*<sup>1</sup> (ToF). Ein ToF ist ein Entfernungsmesser, der auf der Laufzeitmessung eines ausgesendeten Lichtsignals beruht. Auf Basis der Echosignallaufzeit des ausgesendeten Signals lässt sich die Distanz zwischen Sensor und reflektierendem Objekt bestimmen. Dadurch ist eine berührungslose Entfernungsmessung mit einem einzelnen Sensor möglich. Bei einer angenommenen Lichtgeschwindigkeit von  $c = 3 \cdot 10^6 \frac{m}{s}$  und einer gemessenen Echosignallaufzeit  $t_{echo}$  ist die Distanz  $r$  zwischen Sensor und Objekt:

$$r = c \cdot 0,5 \cdot t_{echo} \quad (2.1)$$

ToF-Sensoren gewinnen in den letzten Jahren zunehmend an Bedeutung in der 3D-Umfeldwahrnehmung.[1] Gegenüber Stereo-Kamerasystemen, die auch zur 3D-Umfeldwahrnehmung eingesetzt werden, besitzen sie zahlreiche Vorteile. Dazu zählt, dass ToF-Sensoren nicht auf Textur der beobachteten Szene angewiesen sind, die bei Stereokamerasystemen zur korrespondenzbasierten Tiefenschätzung zwingend in einer Szene enthalten sein muss. Zudem kommen ToF-Sensoren mit einem einzelnen Sensor statt zweier Sensoren aus und messen die Entfernung direkt, ohne dass weitere komplexe Berechnungen wie bei der Stereoskopie erforderlich sind.

Die in dieser Arbeit eingesetzte ToF-Kamera ist von *PMD Technologies* und ist ein sogenannter Photonenmischdetektor (PMD). In einem PMD wird die Lichtlaufzeit nicht direkt gemessen, sondern über eine Phasenverschiebung zwischen dem ausgesendeten und dem empfangenen modulierten Signal, da die direkte Messung der Laufzeit enorm hohe Anforderungen an die Genauigkeit der Signallaufzeitmessung stellt. Gemäß Formel 2.1 ist für eine Entfernungsauflösung von 1 cm eine Genauigkeit der Signallaufzeitmessung von  $66,67 \cdot 10^{-12}$  s erforderlich.

---

<sup>1</sup> *Time-of-Flight*, engl.: „Laufzeit“

Ist jedoch das Ausgangssignal mit bekannter Modulationsfrequenz moduliert, besteht (unter Vernachlässigung von Mehrdeutigkeiten) ein linearer Zusammenhang zwischen der Phasendifferenz  $\varphi$  und der Entfernung  $r$  zwischen Sensor und Objekt:

$$r = \frac{c}{2 \cdot f_{mod}} \cdot \frac{\varphi}{\pi} \quad (2.2)$$

$c$  ist die Lichtgeschwindigkeit,  $f_{mod}$  die Modulationsfrequenz des Signals und  $\varphi$  die gemessene Phasendifferenz. Die Modulationsfrequenz gibt damit die Reichweite des PMD-Sensors vor, da periodisch im Abstand der Wellenlänge Mehrdeutigkeiten auftreten.

Die Messung der Phasendifferenz beruht auf einer gleich getakteten Modulation des Senders und des Empfängers. Die Modulation der sendenden Infrarotdioden ist synchron mit einem erzeugten Potentialgefälle auf der Empfängersseite. Dieses Potentialgefälle dient dazu, die durch das auftreffende Licht freigesetzten Elektronen im Wechsel auf jeweils eine von zwei Auslesedioden abzulenken. Eine solche Potentialwippe mitsamt der Auslesedioden und -elektronik stellt einen Pixel des PMD-Sensors dar. Abbildung 2.2 zeigt einen solchen Pixel. Die Kombination mehrerer Pixel bildet den Sensor einer 3D-Kamera.

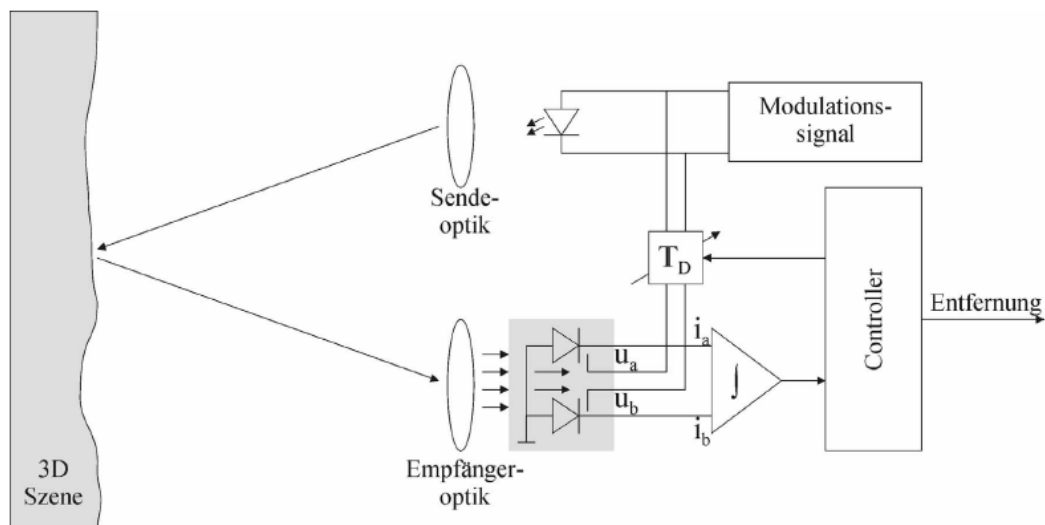


Abbildung 2.2: Schema eines Pixels der *PMD[vision] O3*. [2]

Trifft nicht moduliertes (Umgebungs-) Licht auf einen PMD-Pixel, nehmen beide Auslesedioden gleich viele Elektronen auf. Trifft moduliertes Licht auf, so werden durch die synchrone Modulation des Senders und Empfängers die Elektronen den beiden Auslesedioden in unterschiedlicher Menge zugeführt. Aus der Differenz der in den beiden Auslesedioden gemessenen Elektronen lässt sich so unter Vernachlässigung des Umgebungslichts auf die Phasendifferenz zwischen gesendetem und empfangenem Signal und damit gemäß Formel 2.2 auf die Distanz zwischen PMD-Sensor und Objekt schließen.

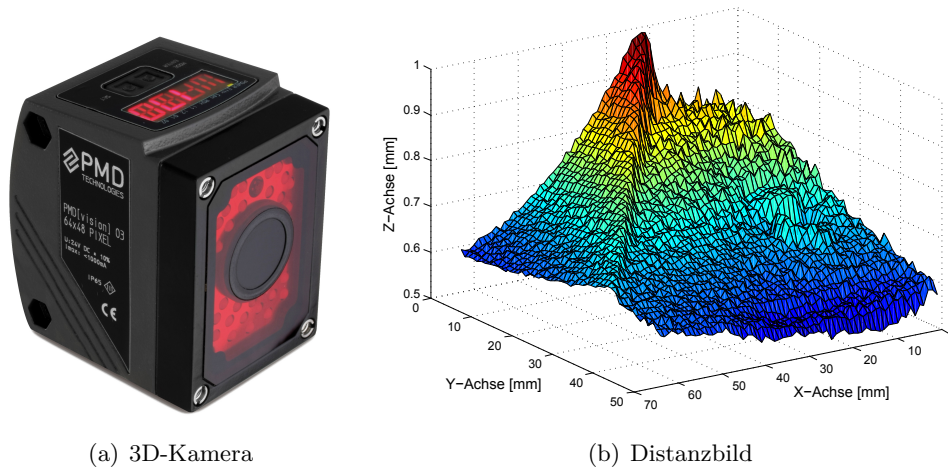


Abbildung 2.3: (a) Die in dieser Arbeit eingesetzte 3D-Kamera *PMD[vision] O3* [PMD Tech] (b) Distanzbild dargestellt als Oberfläche.

### PMD[vision] O3

Der in dieser Arbeit benutzte Sensor *PMD[vision] O3* (siehe Abbildung 2.3(a)) hat eine Auflösung von  $64 \times 50$  Pixel und arbeitet mit einer Modulationsfrequenz von  $20 \text{ MHz}$ . Abbildung 2.3(b) zeigt eine Beispielaufnahme der 3D-Kamera, in der die gemessenen Distanzen der einzelnen Pixel als Oberfläche dargestellt sind.

Da das Signal die Distanz zwischen Sensor und Objekt zweimal zurücklegt, beträgt die eindeutig messbare maximale Objektdistanz  $r_{max}$  die halbe Wellenlänge des modulierten Signals:

$$r_{max} = 0,5 \cdot \frac{c}{f_{mod}} = 0,5 \cdot \frac{3 \cdot 10^8 \frac{\text{m}}{\text{s}}}{20 \cdot 10^6 \frac{1}{\text{s}}} = 7,5 \text{ m} \quad (2.3)$$

Neuere Modelle der PMD-Sensoren lösen das Problem des eingeschränkten Eindeutigkeitsbereichs über den Wechsel zwischen zwei unterschiedlichen Modulationsfrequenzen. Durch die beiden unterschiedliche Eindeutigkeitsbereiche der beiden Frequenzen lässt sich der eindeutige Messbereich auf über 100 Meter erweitern. Die Begrenzung der Reichweite ist dann durch die Stärke der Lichtquelle, nicht mehr durch die Mehrdeutigkeit, gegeben.[2]

Die Messung mit PMD-Sensoren leidet unter den Einflüssen einiger systematischer Fehler. Diese Fehler lassen sich nicht allein als konstanter Offset der gemessenen Distanz abbilden, sondern beschreiben einer mehrdimensionale Funktion. Zur Modellierung des Fehlers steht eine Toolbox zur Verfügung.[3]

#### 2.1.3 Omnidirektionale Kamera

Übliche monokulare Kameras haben abhängig von der Brennweite der Optik ein eingeschränktes Sichtfeld mit Öffnungswinkeln von typischerweise 30 bis 50 Grad. Der dadurch abgedeckte

Sichtbereich ist für einige Anwendungen der Robotik nicht ausreichend. Für Applikationen wie zum Beispiel Lokalisation und Navigation ist ein größeres Sichtfeld vorteilhaft oder notwendig. Eine mögliche Lösung für größere Sichtbereiche stellen omnidirektionale Kamerasysteme dar. Omnidirektionale Kamerasysteme nehmen Bilder einer 360-Grad-Rundumsicht auf.<sup>2</sup> Üblicherweise eine (mitunter nicht vollständige) nach oben oder unten geöffnete Hemisphäre oder einen zylinderförmigen Bereich rund um die die Kamera.

Systeme, die dies ermöglichen, sind zum Beispiel eine Zusammenschaltung mehrerer unterschiedlich orientierter Kameras, Kameras mit extremen Weitwinkel-Objektiven<sup>3</sup> oder katadioptrische Systeme. Katadioptrische Systeme sind optische Systeme bestehend aus einer Kombination von optischen Linsen und Spiegeln.<sup>4</sup> Abbildung 2.4 zeigt das in dieser Arbeit verwendete katadioptrische System und eine einzelne Aufnahme der Kamera.

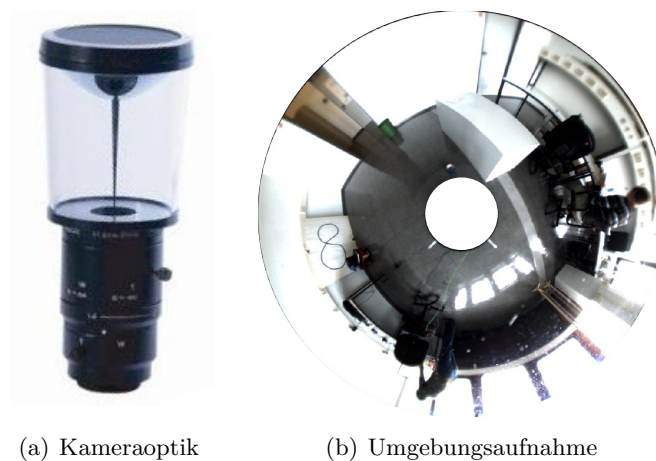


Abbildung 2.4: Katadioptrische Kameraoptik.

Omnidirektionale Kameras, insbesondere katadioptrische Systeme, finden in den letzten Jahren zunehmende Verbreitung in Anwendungen des maschinellen Sehens und vor allem der mobilen Robotik.[4]

Bei den Spiegeln katadioptrischer Systeme handelt es sich um gewölbte Spiegel, die einfallendes Licht aus allen Richtungen der horizontalen Ebene und eines vertikalen Öffnungswinkels auf den Sensor der Kamera lenken. So kann die Kamera eine Rundumansicht ihrer Umgebung aufzeichnen.

<sup>2</sup>Ursprünglich bezeichnet *omnidirektional* die Fähigkeit, 360 Grad horizontal *und* vertikal, also die komplette Sphäre abzubilden („omni“ von lateinisch *omnis*: „jeder“, „ganz“).

<sup>3</sup>Weitwinkel-Objektive, auch *Fischaugen*-Objektive genannt, sind spezielle Objektive mit geringer Brennweite im Vergleich zur Größe des Bildformates. Sie haben dadurch ein sehr viel größeres Sichtfeld als normale Objektive.

<sup>4</sup>*Dioptrik* ist die Wissenschaft der brechenden Systeme (Linsen), *Katoptrik* ist die Wissenschaft der reflektierenden Systeme (Spiegel). *Katadioptrik* bezeichnet die Kombination von Linsen und Spiegeln in einem System.

Ein wichtiges Kriterium für omnidirektionale Kamerasysteme ist für viele Anwendungen die Beschränkung auf einen einzelnen effektiven Blickpunkt (*Single Viewpoint Constraint, SVC*). Diese Einschränkung besagt, dass das System nur Licht aufnimmt, das durch einen einzelnen, virtuellen Punkt im dreidimensionalen Raum fällt. Dieser Punkt wird als effektiver Blickpunkt (*Effective Viewpoint*) bezeichnet. Der Bildsensor bildet damit die 5-dimensionale plenoptische Funktion dieses Raumpunktes ab.[4]

Vorteil einer Systems, das diese Bedingung erfüllt, ist, dass perspektivisch korrekte Bilder erzeugt werden und damit auch eine perspektivisch korrekte Rekonstruktion von Panorama-Bildern möglich ist.[5] Dadurch wird eine Sensorfusion mit weiteren, perspektivischen Kameras, wie zum Beispiel monokularen Kameras, möglich. Darüber hinaus können sämtliche Algorithmen der Bildverarbeitung, die perspektivische Ansichten voraussetzen, auf die Abbildungen der Systeme übertragen werden.

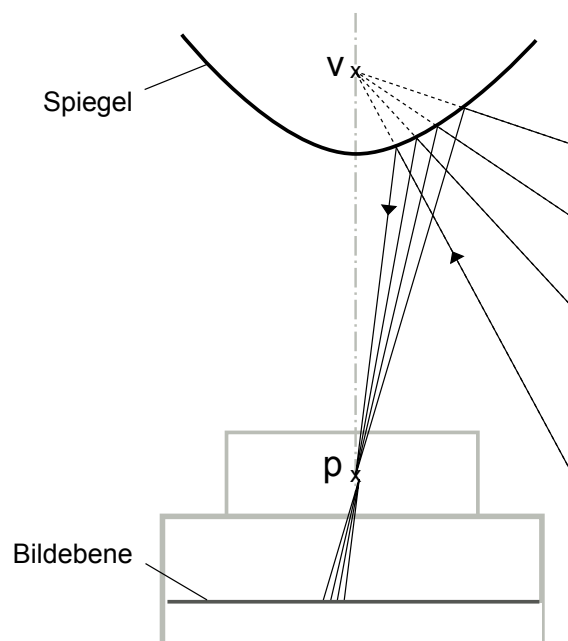


Abbildung 2.5: Lichteinfall in einem katadioptrischen System mit Hyperbolspiegel.[6] Zu sehen ist, dass das einfallende Licht, das den Brennpunkt  $p$  der Kamera passiert, Lichtstrahlen abbildet, die auf den effektiven Blickpunkt  $v$  zulaufen.

Mögliche Formen für ein katadioptrisches System, das den SVC erfüllt, sind perspektivische Kameras in Kombination mit ellipsoiden oder hyperboloiden Spiegeln.[4] Abbildung 2.5 zeigt ein katadioptrisches System mit hyperbolischem Spiegel, wie es in dieser Arbeit angewendet wird. Ausführliche Betrachtungen zum SVC und Systemen, die es erfüllen, finden sich bei Baker und Nayar[4].



### Konstruktion der Kameralhalterung

Eine omnidirektionale Kamera auf einem mobilen Roboter bietet Vereinfachungen, wenn ihre optische Achse der Rotationsachse des Roboters entspricht. Ist dies der Fall, führt eine Rotation des Roboters zu einer reinen Rotation der Kamera um ihre optische Achse. Entspricht die optische Achse der Kamera nicht der Rotationsachse des Roboters, führt eine Rotation des Roboters zu einer Rotation und Translation der Kamera. Insbesondere für die bildbasierte Regelung ist dies interessant, da Rotation und Translation des Roboters und der Kamera/des Kamerabildes entkoppelt werden. Die folgende Konstruktion berücksichtigt dies und platziert die omnidirektionale Kamera in der Rotationsachse des mobilen Roboters.

Zu Beginn der Arbeit stand eine omnidirektionale Kamera zur Verfügung, die einen Öffnungswinkel von 5 Grad unter dem Horizont und 60 Grad über dem Horizont aufweist. Der Sichtbereich der Kamera ist damit für eine Segmentierung des Bodens, wie sie für die vorliegende Aufgabenstellung notwendig ist, nicht geeignet. Um den Boden zu segmentieren wurde daher eine Konstruktion entworfen, mit der die Kamera den Boden abbilden kann, ohne in ihrem Sichtbereich nennenswert beschränkt zu werden. Die Kamera wurde dafür mitsamt des katadioptrischen Systems kopfüber montiert.

Bei der Konstruktion wurde zum einen darauf Wert gelegt, den Sichtbereich der Kamera möglichst wenig durch die Konstruktion selbst oder Kabel zu verdecken und zum anderen die Konstruktion unempfindlich gegenüber Bewegungen des Roboters zu gestalten, damit trotz des hohen Schwerpunktes der Kamera ein scharfes Umgebungsbild aufgenommen werden kann. Abbildung 2.6 zeigt die Konstruktion bestehend aus drei Metallblechen sowie den verdeckten Sichtbereich im Bild.

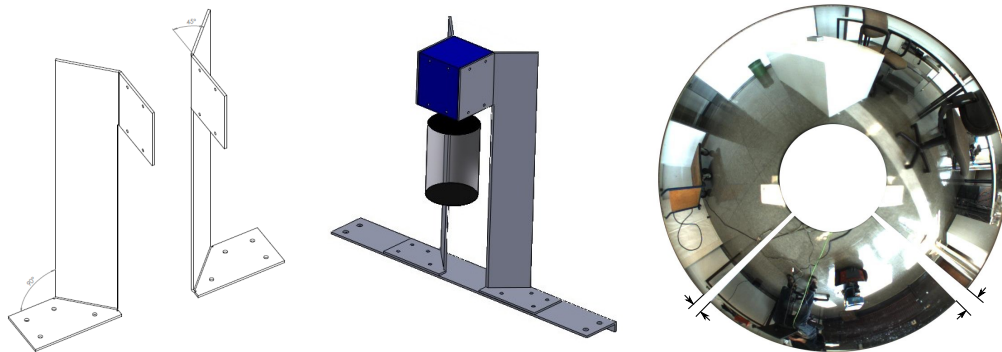


Abbildung 2.6: Links und Mitte: Konstruktion mit der omnidirektionalen Kamera. Rechts: Durch die Konstruktion eingeschränkte Sichtbereiche der omnidirektionalen Ansicht.

Dadurch, dass die Kamera auf die schmale Seite der vertikalen Bleche blickt, ist die Einschränkung des Sichtfeldes minimal. Der verdeckte Bildbereich kann zudem einfach im Rahmen der Bildverarbeitung automatisch entfernt werden. Das Datenkabel der Kamera kann für die Kamera unsichtbar auf der Rückseite der Bleche geführt werden. Dadurch, dass die Bleche im 90-Grad-Winkel zueinander stehen, ist die Konstruktion hinreichend verwindungssteif und stabil.

Im Verlauf der Arbeit mit dem Experimentalsystem wurde die Optik durch eine neue Version ersetzt, die einen vertikalen Öffnungswinkel von 15 Grad über dem Horizont und 60 Grad unter dem Horizont aufweist. Die neue Optik ist damit nicht auf die beschriebene Konstruktion angewiesen und zeigt eine Umgebungsansicht ohne Einschränkung des Sichtbereichs durch die Blechkonstruktion oder Kabel.

## 2.2 Kamerakalibrierung

Kamerakalibrierung bezeichnet das Ermitteln desjenigen Parametersatzes, mit dem sich eine Kamera oder ein System mehrerer Kameras mathematisch beschreiben lässt. Dabei wird zwischen intrinsischer und extrinsischer Kalibrierung unterschieden. Die intrinsische Kalibrierung ermittelt die intrinsischen Parameter der Kameras (Sensorgeometrie, Brennweite, Verzerrung), die extrinsische Kalibrierung ermittelt Orientierung und Position der Kamera beziehungsweise der Kameras zueinander.

Sind sowohl intrinsische als auch extrinsische Kameraparameter bekannt, bezeichnet man die Kamera als vollständig kalibriert. Ist eine Kamera vollständig kalibriert, kann die Projektion eines beliebigen Punktes im Raum auf die Bildebene der Kamera berechnet werden. Umgekehrt kann ein Punkt auf der Bildebene bis auf die Tiefe im Bild in einen 3-dimensionalen Punkt umgerechnet werden.

Sind zwei Kameras kalibriert, beschreibt die Epipolargeometrie die Beziehung zwischen Abbildungen eines einzelnen dreidimensionalen Punktes auf zwei Kameras. Über die epipolare Beziehung ist zu einem Bildpunkt in dem einen Kamerabild eine Linie (die Epipolarlinie) bekannt, auf dem dort der Bildpunkt liegt, der den gleichen Raumpunkt abbildet.

Mit der in dieser Arbeit verwendeten ToF-Kamera ist jedoch anders als bei üblichen Kameras auch die Tiefe der Raumpunktes bekannt. Sind ToF-Kamera und eine weitere Kamera vollständig kalibriert, lässt sich somit zu einem Bildpunkt der ToF-Kamera der korrespondierende Bildpunkt der weiteren Kamera exakt bestimmen.

### Tiefenkalibrierung

Wie in Kapitel 2.1.2 erwähnt, unterliegen die Aufnahmen des PMD-Sensors einer Reihe von systematischen Fehlern. Um diese in das Sensormodell mit einzubeziehen und damit die Qualität der Messung zu verbessern, lässt sich eine sogenannte Tiefenkalibrierung durchführen. Die Tiefenkalibrierung korrigiert die Messung abhängig von der gemessenen Tiefe und der Position im Bild.[3]

Die Kalibrierungs-Toolbox von Schiller et al.[3] korrigiert die Tiefeninformation über einen Satz von Parametern  $d_0, d_1, \dots, d_5$ , der über die Kalibrierung bestimmt werden kann. Der korrigierte Tiefenwert  $\lambda^*$  errechnet sich aus dem gemessenen Tiefenwert  $\lambda$  anhand folgender Funktion:

$$\lambda^* = d_0 + (1, 0 + d_1) \cdot \lambda + d_2 \cdot u + d_3 \cdot v + d_4 \cdot \lambda^2 + d_5 \cdot \lambda^3 \quad (2.4)$$

Die Variablen  $u$  und  $v$  sind die Pixelkoordinaten.

## 2.3 Überwachtes maschinelles Lernen

Im Rahmen der Arbeit wird ein Roboterverhalten mit Hilfe maschinellen Lernens aufgrund aufgezeichneter Trainingsdaten erzeugt. Durch die Trainingsdaten ist bereits das Ziel des Lernens für einen Satz von Daten bekannt. Findet maschinelles Lernen auf dieser Basis statt, spricht man von überwachtem Lernen. Überwachtes Lernen ist ein Teilgebiet des maschinellen Lernens, bei dem im Gegensatz zum unüberwachten Lernen vor dem Prozess bereits die Ergebnisse zu einem Satz von Trainingsfällen bekannt sind. Überwachtes Lernen versucht dann anhand der Trainingsfälle und der zugehörigen bekannten Ergebnisse Gesetzmäßigkeiten zu finden, um zukünftige Probleme selbstständig zu lösen.

Überwachtes Lernen kann als Modellbildung des Zusammenhangs zwischen Eingangs- und Ausgangsgrößen betrachtet werden. Für überwachtes Lernen steht als Trainingsdaten ein sogenannter Lernvektor (*Teaching Vector*) als Satz von Eingabewerten und bekannten, richtigen Ausgabewerten zur Verfügung. Eingabe- und Ausgabegröße sind im Allgemeinen mehrdimensional.

Die Modellbildung erfolgt anhand des Lernvektors in einer Form, die die Voraussagen des Modells den bekannten Ergebnissen möglichst nahe kommen lässt. Grundlage des Vergleichs der Voraussage des Systems und den bekannten Ergebnissen ist ein definiertes Fehlermaß. Dieses Fehlermaß beschreibt die *Ähnlichkeit* zweier Lösungen. Durch den Lernprozess soll dieses Fehlermaß zwischen bekannter, richtiger Lösung und gelernter Lösung minimiert werden. Ein mögliches Fehlermaß ist zum Beispiel der mittlere quadratische Fehler.

Ziel eines angelernten/trainierten Systems ist es, zu einem unbekanntem Problem der trainierten Problemklasse korrekte Lösungen zu liefern. Dies kann mithilfe eines Trainingssets und eines Testsets validiert werden. Dabei wird das System mit dem Trainingsset trainiert und anschließend das Fehlermaß auf dem Testset bestimmt.

Im Folgenden werden zwei Verfahren überwachtem Lernen vorgestellt, die für diese Arbeit auf ihre Eignung untersucht wurden.

### 2.3.1 Künstliche neuronale Netze

Künstliche neuronale Netze (KNN) stellen eine Möglichkeit des überwachtem Lernens dar. KNN sind Netze aus künstlichen Neuronen und damit von biologischen neuronalen Netzen, wie etwa dem menschlichen Gehirn, inspiriert.

KNN basieren auf der Vernetzung vieler künstlicher Neuronen durch Synapsen. Ein künstliches Neuron ist eine Funktion seiner Eingangswerte. Jedes Neuron summiert dabei alle Ein-

gangswerte und erzeugt auf dieser Basis einen Ausgabewert. Häufig wird dabei als Ausgabewert eine 1 ausgegeben, falls die Summe der Eingangswerte eine bestimmten Schwelle  $\Theta$  überschreiten, sonst gibt das Neuron eine 0 zurück. Die Eingänge der Neuronen sind dabei in der Regel gewichtet.

Das KNN entsteht durch Zusammenschaltung der Neuronen über Synapsen (gewichtete Verbindungen), so dass die Ausgangswerte der Neuronen die gewichteten Eingangswerte weiterer Neuronen darstellen.

Beim überwachten Lernen wird dem KNN ein Eingangsmuster gegeben und die Ausgabe, die das KNN in seinem aktuellen Zustand produziert, mit dem bekannten richtigen Ausgabewert verglichen. Durch diesen Vergleich von Soll- und Istausgabe kann auf die vorzunehmenden Änderungen der Netzkonfiguration geschlossen werden. Die Änderung des Netzes erfolgt in der Regel durch Anpassung der Gewichte der Neuroneneingänge und der Schwellwerte der Neuronen, kann aber prinzipiell auf vier Arten geschehen:

- Entwicklung neuer Verbindungen, Löschen bestehender Verbindungen
- Ändern der Gewichtung (der Gewichte  $w_{ij}$  von Neuron  $i$  zu Neuron  $j$ )
- Anpassen der Schwellwerte der Neuronen
- Hinzufügen oder Löschen von Neuronen

KNN sind in der Lage, komplizierte, nichtlineare Zusammenhänge über einen Lernalgorithmus zu modellieren.

### 2.3.2 Instanzbasiertes Lernen

Instanzbasiertes Lernen bezeichnet das Lernen aufgrund einer Wissensdatenbank bisheriger Probleme und deren Lösungen (in diesem Kontext „*Instanzen*“ eines Problems genannt). Um die *Klasse* eines aktuellen Problems vorherzusagen, wird es mit gespeicherten *Instanzen* voriger Probleme, deren Klasse bekannt ist, verglichen.

Anders als bei vielen anderen Lernverfahren wird dabei nicht versucht, aus den Beispielen einer Klasse von Problemen und deren Lösungen eine Abstraktion oder Generalisierung vorzukompilieren, sondern es wird direkt in den gespeicherten Instanzen die Lösung zu der aktuellen Probleminstanz *gesucht*. Die Lösung eines Problems basiert damit immer auf bereits als funktionierend bekannten Lösungen.

Eine Instanz ist ein Attribut-Wert-Paar bestehend aus dem  $n$ -dimensionalen Merkmalsvektor  $\mathbf{x} = (x_1, x_2, \dots, x_n)$ , und der Ausgangsgröße  $\mathbf{y}$ .  $x_1 - x_n$  sind die Attribute/Merkmale des Problems, die Ausgangsgröße ist zum Beispiel die Problemklasse.

Der Merkmalsvektor spannt einen  $n$ -dimensionalen *Instanzraum* auf, in dem ein Abstandmaß  $d(\mathbf{x}^1, \mathbf{x}^2)$  definiert ist, das den Abstand zweier Instanzen  $\mathbf{x}^1$  und  $\mathbf{x}^2$  angibt. Typischerweise ist dies der euklidische Abstand

$$d(\mathbf{x}^1, \mathbf{x}^2) = \sqrt{\sum_{i=1}^n (x_i^1 - x_i^2)^2} \quad (2.5)$$

oder die Manhattan-Distanz

$$d(\mathbf{x}^1, \mathbf{x}^2) = \sum_{i=1}^n |x_i^1 - x_i^2| \quad (2.6)$$

Das Abstandsmaß ist ideal, wenn Instanzen gleicher Klasse einen geringen Abstand aufweisen und Instanzen unterschiedlicher Klassen einen großen Abstand aufweisen.

Während viele andere Lernverfahren im Zuge des Lernprozesses eine Beispielmenge analysieren und eine generelle Hypothese erzeugen, besteht die Lernphase bei instanzbasierten Verfahren lediglich aus dem Abspeichern der Instanzen (Merkmalsvektor  $\mathbf{x}$  und Klassifizierung  $y$ ). Der eigentliche Rechenaufwand fällt dadurch nicht beim Trainieren an, sondern in Form einer Suche im Merkmalsraum erst zum Zeitpunkt der konkreten Problemstellung. Instanzbasiertes Lernen wird daher auch als *Lazy Learning* (Träges Lernen) bezeichnet.

### Lazy Learning

Prinzipiell kann bei maschinellem Lernen zwischen *Lazy Learning* und *Eager Learning* (Eifriges Lernen) unterschieden werden. Während beim *Eager Learning* während des Trainierens des Systems bereits das Modell gelernt wird, findet beim *Lazy Learning* die Modellbildung, also das maschinelle Lernen, erst zur Zeit der Abfrage statt.[7]

Ein großer Vorteil von *Lazy Learning* ist, dass zur Zeit der Problemstellung die Modellbildung lokal, also in der direkter Umgebung des aktuellen Arbeitspunktes, geschehen kann. Beim *Eager Learning* findet die Modellbildung stets vorher, ohne Kenntnis der konkreten Problemstellung, und daher immer global statt. Abbildung 2.7 zeigt, wie dadurch aufgrund der lokalen Umgebung ein genaueres Modell gelernt werden kann.

Abseits der Vorteile besitzt *Lazy Learning* nachteilige Eigenschaften. Diese sind zum Beispiel:[8, S. 39]

1. Der Rechenaufwand fällt zur Zeit der Problemstellung an
2. Das Ergebnis ist empfindlich gegenüber Rauschen
3. Das Ergebnis ist empfindlich gegenüber irrelevanten Attributen

Diese Probleme müssen bedacht und berücksichtigt werden, um mit *Lazy Learning* zu vernünftigen Ergebnissen zu finden.

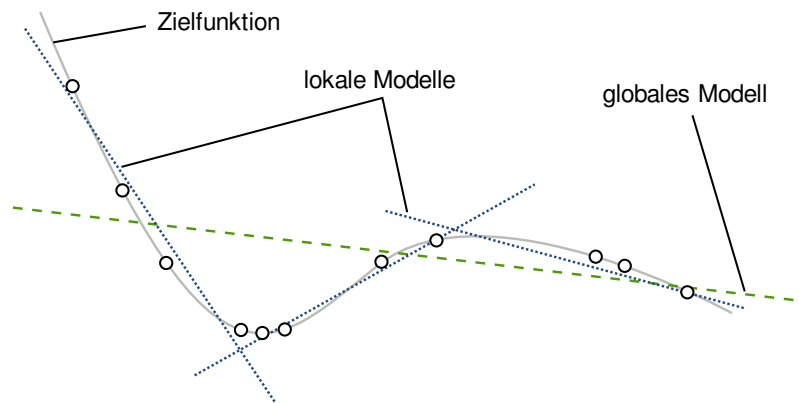


Abbildung 2.7: Unterschied zwischen *Lazy Learning* und *Eager Learning* im zweidimensionalen, kontinuierlichen Fall. Mit *Lazy Learning* kann bei jeder Abfrage das lokale, genauere Modell (gepunktet) gelernt werden, beim *Eager Learning* nur das globale, ungenauere (gestrichelt).

### Klassifizierung

Die Klassifikation eines Objekts  $\mathbf{x}_q$  mittels instanzbasierten Lernens erfolgt im einfachsten Fall durch Mehrheitsentscheidung unter den nächsten Nachbarn des zu klassifizierenden Objektes (Nächste-Nachbarn-Klassifikation, *K-Nearest-Neighbour*). Dazu werden zunächst die  $k$  nächsten Nachbarn zum Abfragepunkt  $\mathbf{x}_q$  bestimmt. Die Klasse der  $k$  nächsten Nachbarn ist bekannt, dadurch kann über Mehrheitsentscheidung die Klasse der Abfragepunktes bestimmt werden. Abbildung 2.8 zeigt, wie die Wahl des Parameters  $k$  die Klassifizierung beeinflusst.

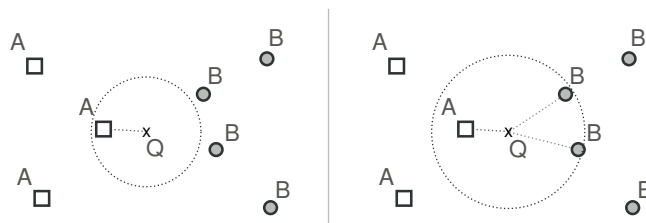


Abbildung 2.8: Klassifizierung über den Nächste-Nachbarn-Klassifikator. Im Beispiel würde der alleinige Nachbar dem Abfragepunkt  $\mathbf{x}_q$  die Problemklasse A zuordnen, die  $k$  nächsten Nachbarn (mit  $k = 3$ ) klassifizieren den Punkt als B.

### Regression

Im Unterschied zur Klassifizierung sagt die Regression keine diskreten Werte, sondern eine kontinuierliche Funktion  $f(\mathbf{x})$  voraus. Der einem Merkmalsvektor zugeordnete Ausgabewert  $y$  einer Instanz ist dementsprechend eine reelle Zahl.

Im Gegensatz zur Klassifizierung findet dabei unter den  $k$  nächsten Nachbarn keine Mehrheitsentscheidung statt, sondern es wird aus den umgebenden  $k$  Nachbarn ein kontinuierlicher Wert erzeugt. Dies kann zum Beispiel durch Mittelung der Ausgabewerte  $y^j$  (mit  $j = 1, 2, \dots, k$ ) der  $k$  nächsten Nachbarn erfolgen. Häufig verwendet ist auch das distanz-gewichtete Mittel (*Distance Weighted Average*, DWA), bei dem bei der Mittelung die einzelnen Nachbarn anhand ihrer Distanz gewichtet werden, so dass Instanzen mit höherer Distanz zum Abfragepunkt  $\mathbf{x}_q$  schwächer beitragen.

### 2.3.3 Vergleich der Lernverfahren

Der Einsatz von künstlichen neuronalen Netzen zum maschinellen Lernen von Roboterverhalten bietet den Vorteil, dass der Lernaufwand zur Trainingszeit, nicht zur Abfragezeit anfällt (*Eager Learning*). Der Rechenaufwand zur Laufzeit des Programms ist damit gering. Da zum Zeitpunkt des Lernens der Abfragepunkt noch nicht bekannt ist, kann das Lernen jedoch nur global, auf der gesamten Datenmenge, stattfinden.

*Lazy Learning* als Verfahren bildet das Modell zum Abfragezeitpunkt und lokal. Der Vorteil ist, dass damit mehrere Problemklassen zugelassen und modelliert werden können; in der hiesigen Anwendung zum Lernen von Roboterverhalten sich also mehrere verschiedene Roboterverhalten ausbilden können. Die Modellbildung findet darüber hinaus bei *Lazy Learning* transparent statt, so dass zum Abfragezeitpunkt die Modellbildung nachvollzogen werden kann. Fehler in der Ausgabe können dadurch auf die Ursachen zurückverfolgt und leichter behoben werden. Bei künstlichen neuronalen Netzen lassen die gelernten Schwellwerte der Neuronen und Gewichte der Synapsen nur bedingt eine spätere Fehleranalyse zu.

Grundlegender Vorteil von *Lazy Learning* im Kontext der hiesigen Anwendung ist zudem, dass lediglich gespeicherte, bekannte Bewegungen des Roboters wiederholt werden. Eine globale Generalisierung, die verschiedene Bewegungen des Roboters generalisiert und dabei möglicherweise eine fehlerhafte neue Bewegung als Lösung erzeugt, findet nur bedingt, nämlich lediglich unter den nächsten Nachbarn der Abfrageinstanz statt.

Die Möglichkeit der lokalen Modellbildung und der Nachvollziehbarkeit bei der Modellbildung von *Lazy Learning*, sowie vor allem die Vermeidung der Erzeugung von Bewegungen, die nicht trainiert sind, wird für diese Arbeit der höheren Performance von künstlichen neuronalen Netzen zum Abfragezeitpunkt vorgezogen. Die in Kapitel 2.3.2 aufgezeigten Nachteile von *Lazy Learning* müssen jedoch beim Lernen des Roboterverhaltens berücksichtigt werden.

#### Lazy Learning Toolbox

In dieser Arbeit wird für das maschinelle Lernen die *Lazy Learning Toolbox* für MatLab von Bontempi[9] benutzt. Die Toolbox ermittelt die Problemklasse über die Auswahl der  $k$  nächsten Nachbarn zum Abfragepunkt.

Die Anzahl  $k$  der nächsten Nachbarn kann konkret oder in einer Spanne vorgegeben werden. Zur Regression erlaubt die Toolbox die Wahl verschiedener lokaler Modelle. Zur Auswahl steht dafür ein konstantes Modell, ein lineares Modell, ein quadratisches Modell sowie die

automatische Auswahl des bestmöglichen Modells zum Abfragezeitpunkt. Bei der automatischen Auswahl des besten Modells werden bei einer Abfrage unterschiedliche Anzahlen von nächsten Nachbarn, die verschiedenen Modelle sowie Kombination der Modelle getestet und die beste Kombination für die Antwort des Systems verwendet.[10]



## Kapitel 3

# Wahrnehmung

Grundlage für das maschinelle Lernen des Roboterverhaltens ist eine für die Aufgabe geeignete Repräsentation der Umgebung und, dem zugrunde liegend, die Wahrnehmung des Umfeldes. Durch die Wahrnehmung und Repräsentation der Umgebung wird der Roboter in die Lage versetzt, die aktuelle Situation zu bewerten und anhand dessen selbstständig Entscheidungen über folgenden Aktionen zu treffen.

Die im *Pioneer 3-DX* verbauten Sonar-Sensoren des Roboters kommen in der vorliegenden Arbeit nicht zum Einsatz. Stattdessen basiert die Wahrnehmung ausschließlich aus drei bildgebenden Sensoren: Eine 3D-Kamera, eine monokulare Kamera und eine omnidirektionale Kamera. Ziel ist die Untersuchung, inwieweit sich bildgebende Sensorik, ohne den Einsatz weiterer Sensorik wie RADAR, LIDAR oder Sonar, für eine robuste Roboternavigation einsetzen lässt.

Dies ist von Interesse, da bildgebende Sensoren anders als viele andere Sensoren einen Massenmarkt darstellen und dadurch gegenüber anderen Sensoren einen Kostenvorteil aufweisen. So sind Kosten von 3D-Laserscannern im vier- bis fünfstelligen Bereich, währenddessen Kameras nur einen Bruchteil dieser Kosten ausmachen. Aktuelle 3D-Kameras mit mehreren tausend Pixeln Auflösung, wie sie hier benutzt werden, stehen zwar zur Zeit Laserscannern preislich nicht erheblich nach. Da sie allerdings wie normale Kameras in CMOS-Technik gefertigt werden, ist in den nächsten Jahren eine Annäherung der Fertigungskosten an die von handelsüblichen Kameras zu erwarten.

Mit dem Ziel, die bildgebende Sensorik für die Kollisionsvermeidung und das Korridorfolgeverhalten zu nutzen, muss die nähere Umgebung in Fahrtrichtung des Roboters wahrzunehmen und eine lokale, abstrakte Umgebungsrepräsentation erstellt werden.

Für Ersteres, die Kollisionsvermeidung, ist der Bodenbereich unmittelbar vor dem mobilen Roboter in Fahrtrichtung ausschlaggebend, da sich der Roboter nicht quer zu den Antriebsrädern bewegen kann und in der hiesigen Anwendung nicht rückwärts bewegt. Dementsprechend sind PMD-Sensor und monokulare Kamera auf den Boden vor dem Roboter gerichtet.

Die beiden Sensoren sind, wie in Abbildung 2.1 gezeigt, parallel zueinander und in geringem Abstand zueinander angebracht. Das Sichtfeld der beiden Kameras ist damit ähnlich. Das

Blickfeld der beiden Kameras überschneidet sich zudem mit dem Blickfeld der omnidirektionalen Kamera. Abbildung 3.1 zeigt die überlagerten Blickfelder der Kameras. Erkennbar ist die Rundumsicht der omnidirektionalen Kamera, die auf den Boden vor dem Roboter ausgerichtete monokulare und 3D-Kamera, sowie der gemeinsame Blickbereich der drei Kameras.

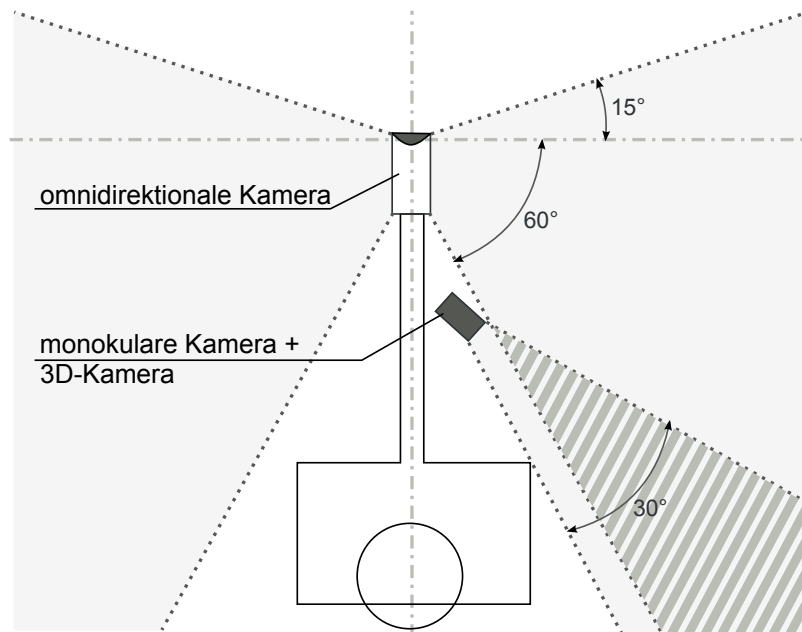


Abbildung 3.1: Blickfelder der drei Kameras. Der schraffierte Bereich zeigt den überlappenden Sichtbereich.

In den überlappenden Bereichen der Kamera-Blickfelder können Korrespondenzen gesucht werden und Informationen der Ansichten fusioniert werden.

Um die Robustheit der Wahrnehmung gegenüber wechselnden Umgebungen und störenden Einflüssen der Bildverarbeitung zu erhöhen, werden die Informationen der 3D-Kamera mit den Informationen der monokularen Kamera und der omnidirektionalen Kamera fusioniert. Dazu werden zuerst die 3D-Informationen verarbeitet und anschließend in die 2D-Ansichten projiziert, um die Bildverarbeitung durch diese Zusatzinformation zu verbessern. In den 2D-Ansichten wird darauf aufbauend das Bild segmentiert, um anschließend in geeignete Merkmale für das Roboterverhalten überführt werden zu können.

Vornehmliches Ziel ist es dafür, in Fahrtrichtung des Roboters und in dessen unmittelbarer Umgebung frei befahrbare Flächen (im Folgenden unter dem Begriff „Boden“ zusammengefasst) zu finden. Diese Bodenfläche dient als Grundlage für die Kollisionsvermeidung und das Korridorfolgeverhalten.

### 3.1 Segmentierung im Raum – 3D

Viele Verfahren der Bildsegmentierung benutzen eine Anfangshypothese als Start für die Bildsegmentierung.[11] Die im Folgenden gezeigte Segmentierung der Messung von 3D-Informationen durch die 3D-Kamera dient als Grundlage für diese Anfangshypothesen. Diese kann darüber hinaus als bekannter, wahrer Wert (*Ground Truth*) zur Bewertung der Segmentierung im Bildbereich verwendet werden.

Die Segmentierung der Abstandinformationen der 3D-Kamera hat zum Ziel, geometrische Objekte zu segmentieren und zu klassifizieren. Dies dient vor allem, aber nicht ausschließlich, dem Finden des geometrischen Objekts „Boden“ als für den Roboter frei befahrbare Fläche.

Die Daten, die die 3D-Kamera zur Verfügung stellt, bilden im Wesentlichen eine Punktwolke im dreidimensionalen Raum. Ein einzelner Punkt dieser Punktwolke ist beschrieben durch die horizontale und vertikale Pixelkoordinate  $u$  beziehungsweise  $v$  der 3D-Kamera, sowie die gemessene Distanz  $r$  zum Objekt. Die Pixelkoordinaten  $u$  und  $v$  bilden ein gleichmäßiges Raster von 64 x 50 Pixeln auf dem Sensor der Kamera. Die Distanz  $r$  reicht von 0 – 750 cm wie in Formel 2.3 dargestellt.

Segmentierung in diesen Sensordaten bedeutet die Zusammenfassung und Zuordnung von mehreren Punkten der Punktwolke zu Objekten. Dies geschieht allein anhand der geometrischen Merkmalen der Punkte (also deren Position und Abstand zueinander). Die Intensitätsinformationen, die den 3D-Punkten zugeordnet sind, werden aufgrund der niedrigen Auflösung der 3D-Kamera zugunsten der hochauflösenden Information der monokularen Kamera ignoriert. Segmentierte Objekte werden anschließend klassifiziert. Klassifizierung bedeutet in diesem Fall, die zusammengefassten geometrischen Objekte anhand ihrer Eigenschaften in die drei Klassen

- Boden
  
- Wand
  
- Hindernis

aufzuteilen.

Um die Punkte dementsprechend zu klassifizieren werden zuerst Ebenen in den Punktwolken gesucht, da alle Punkte, die nicht einer Ebene zugeordnet werden können, als Hindernis klassifiziert werden. Die gefunden Ebenen wiederum lassen sich in die Klassen *Boden* und *Wand* aufteilen lassen.

Um Ebenen in der Punktwolke finden zu können, werden die Punkte zunächst in kartesische Raumkoordinaten transformiert, da die anschließenden Verfahren zur Ebenenfindung im kartesischen Raum operieren.

### 3.1.1 Transformation

Ist der PMD-Sensor intrinsisch kalibriert, können die Punkte der 3D-Punktwolke in ihre kartesischen Raumkoordinaten umgerechnet werden. Die Distanz  $r$  zum Objekt ist aus direkter Messung des PMD-Sensors bekannt, es werden daher die Pixelkoordinaten  $u$  und  $v$  in Azimuth  $\Phi$  und Elevation  $\Theta$  umgerechnet:

$$\begin{aligned}\Phi &= \operatorname{atan}\left(\frac{r}{f_{PMD}} \cdot u\right) \\ \Theta &= \operatorname{atan}\left(\frac{r}{f_{PMD}} \cdot v\right)\end{aligned}\tag{3.1}$$

$f_{PMD}$  ist die Brennweite des PMD-Sensors.

Die Distanz  $r$ , Azimuth  $\Phi$  und Elevation  $\Theta$  entsprechen dreidimensionalen Kugelkoordinaten und können in kartesische Raumkoordinaten umgerechnet werden. Die Umrechnung erfolgt dabei so, dass die  $z$ -Achse in Richtung der optischen Achse des Sensors zeigt und die  $x$ -Achse horizontal verläuft:

$$\begin{aligned}x &= r \cdot \sin \Theta \cdot \cos \Phi \\ y &= r \cdot \cos \Theta \\ z &= r \cdot \sin \Theta \cdot \sin \Phi\end{aligned}\tag{3.2}$$

In der durch die Transformation erzeugten Punktwolke im kartesischen Raum werden nun Ebenen in den Punkten gesucht. Das Finden der Ebenen erfolgt mittels des *RANSAC*-Algorithmus.

### 3.1.2 Finden von Ebenen mittels RANSAC

*RANSAC* (kurz für **R**andom **S**ample **C**onsensus, deutsch etwa "Übereinstimmung mit einer zufälligen Stichprobe,") ist ein Algorithmus, der die Schätzung eines Modells trotz Gegenwart vieler Ausreißer und grober Fehlern innerhalb einer Reihe von Messwerten ermöglicht. Dafür werden aus zufälligen Datenpunkten der Datenmenge Hypothesen erzeugt, die anhand der restlichen Datenpunkte verifiziert werden. Die Hypothese, mit denen die meisten Datenpunkte übereinstimmen, wird verwendet. Hypothesen werden aufgrund einer zufälligen Stichprobe aus der Datenbasis gebildet und enthalten jeweils die Mindestmenge an erforderlichen Datenpunkten zur Erzeugung einer Hypothese.

Aufgrund seiner Robustheit gegenüber Ausreißern wird *RANSAC* häufig im Bereich des maschinellen Sehens eingesetzt. Hier unterstützt *RANSAC* häufig Ausgleichsverfahren wie die Methode der kleinsten Quadrate, die bei einer größeren Anzahl von Ausreißern versagen, durch Berechnung einer um Ausreißer bereinigten Datenmenge, der sogenannten Konsensmenge (*Consensus Set*).

Bezogen auf die Aufgabenstellung, Ebenen in der 3D-Punktwolke zu finden, werden zufällig drei Punkte aus der Punktmenge gezogen, da drei Punkte die Mindestmenge zur Aufstellung einer Ebenen-Hypothese im dreidimensionalen Raum darstellen. Diese drei Punkte werden als Stützpunkte zur Berechnung einer Ebenengleichung verwendet. Die damit erstellte Ebenen-Hypothese wird anschließend auf ihre Übereinstimmung mit der restlichen Punktwolke überprüft. Stimmen hinreichend viele Punkte mit der Ebenen-Hypothese überein, wird diese als wahre Ebene angenommen.

Eine Ebene im Dreidimensionalen besitzt vier Freiheitsgrade. Eine Ebenengleichung kann daher zum Beispiel wie folgt aufgestellt werden:

$$a \cdot x + b \cdot y + c \cdot z + d = 0 \quad (3.3)$$

Die Parameter  $a$ ,  $b$ ,  $c$  und  $d$  bestimmen die Ebene. Dementsprechend ist der Suchraum für das Finden der Ebenen vierdimensional.

Das Finden einer Ebene ist im Kontext von RANSAC das Schätzen des Ebenenmodells aufgrund der 3D-Messung der Kamera. Alle Punkte, die nicht zu einer Ebene gehören, werden dabei als Ausreißer betrachtet. Um nicht nur eine Ebene zu finden, sondern den Boden und alle Wände im Bild zu finden, wird *RANSAC* für das vorliegende Problem mehrfach auf die Punktwolke angewendet. Ist eine Ebene gefunden, werden Punkte dieser Ebene aus der ursprünglichen Punktwolke entfernt und in der verbleibenden Punktmenge (die Ausreißer der vorigen Modellschätzung) wird erneut nach einer Ebene gesucht. Dies wird wiederholt, bis keine Ebene mehr gefunden werden kann, die mindestens eine festgelegte Mindestanzahl von Punkten enthält.

Das Ergebnis dieser wiederholten Anwendung des RANSAC-Algorithmus ist eine Menge von Ebenen sowie die verbleibenden, keiner Ebene zugeordneten Punkte. Die Ebenen des Ergebnisses sind beschrieben durch deren vier Parameter  $a$ ,  $b$ ,  $c$  und  $d$ , sowie den zugeordneten Punkten der Punktwolke. In Abbildung 3.2 ist das Ergebnis einer solchen Iteration aufgezeigt. Oben links ist die Szene aus Sicht der monokularen Kamera zu sehen. Die darauf folgenden Ansichten zeigen jeweils hervorgehoben die Raumpunkte, die Boden, Wand und Hindernis zugeordnet sind.

Die Abbildung simuliert den Blick aus der Frontkamera einer Beispielansicht, mit dem Boden im Vordergrund und den segmentierten und klassifizierten Wänden im Hintergrund. Die durch die Segmentierung erhaltenen Ebenen können anhand ihrer Ebenenparameter klassifiziert werden. So wird im obigen Bild eine Ebene richtig als horizontal und damit als Boden identifiziert. Die übrigen zwei Ebenen werden als vertikal und damit als Wände klassifiziert. Verbleibende Punkte, die nicht den Ebenen zugeordnet werden konnten, werden als Hindernis klassifiziert.

Zu sehen ist, dass einzelne Ausreißer nicht den Ebenen zugeordnet wurden, obwohl sich kein Hindernis im Bild befindet. Diese Ausreißer sind zum einen dem Rauschen der 3D-Kamera geschuldet und zum anderen dem reflektierenden Boden, der vor allem in den oberen Bildbereichen aufgrund des flachen Auftreffens des modulierten Lichts zu fehlerhaften Messpunkten

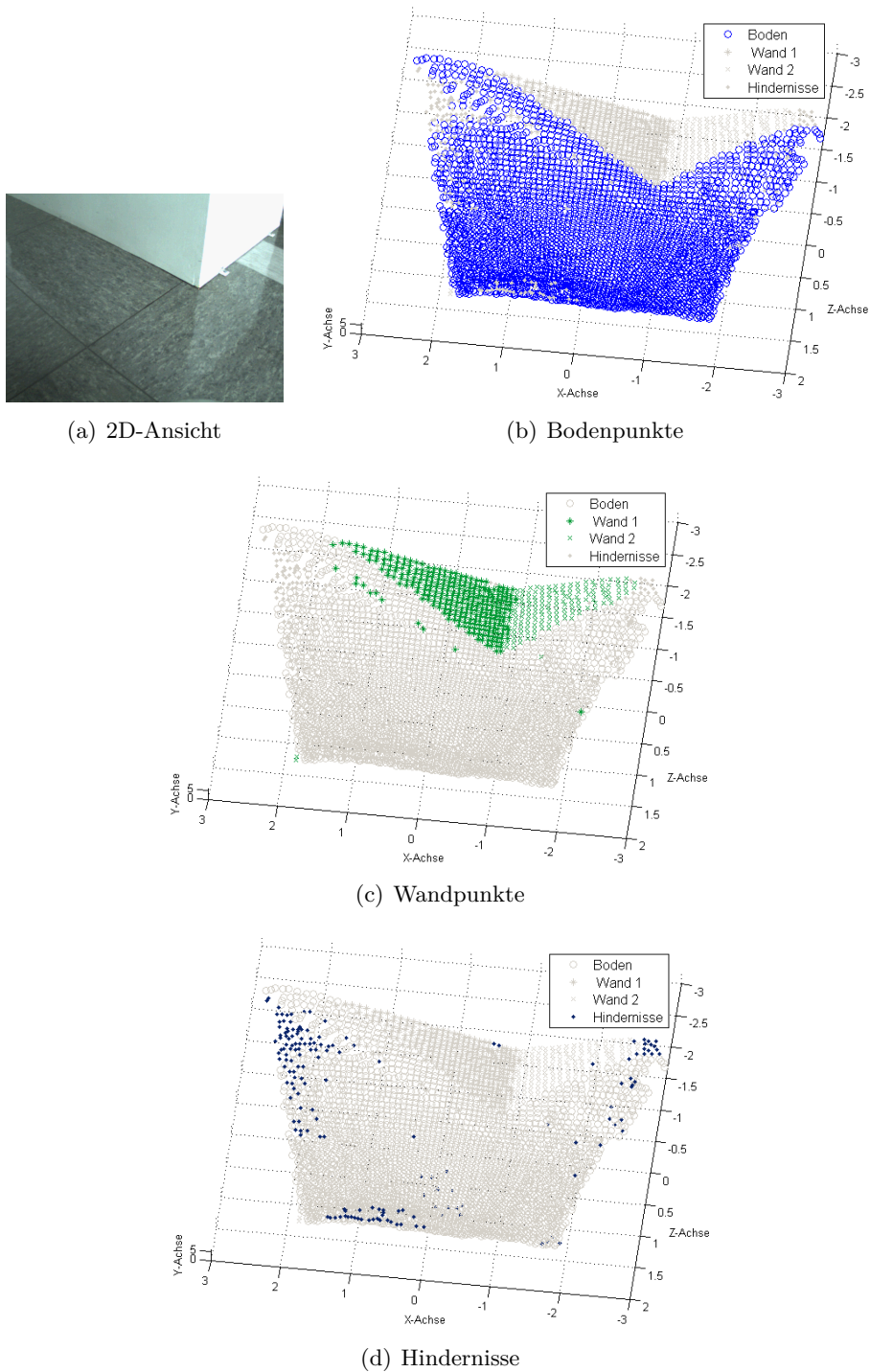


Abbildung 3.2: Segmentierung und Klassifizierung einer 3D-Punktwolke: 2D-Ansicht und Boden-, Wand- und Hindernis-Punkte.

führt.

### 3.1.3 Klassifizierung der Ebenen

Sind die Ebenen gefunden, können sie anhand ihrer Parameter klassifiziert werden. Entsprechend Formel 3.3 erfolgt die Klassifizierung der Ebenen anhand ihrer vier Ebenenparameter.

Die Parameter  $a$ ,  $b$  und  $c$  stellen dabei den Normalenvektor  $\mathbf{n}$  der Ebene dar. Parameter  $d$  stellt den Abstand der Ebene zum Ursprung dar. Der Ursprung der Punktwolke ist der Brennpunkt der 3D-Kamera. Ist die Höhe der Kamera sowie der Nickwinkel wie im Fall dieser Arbeit bekannt, können Wertebereiche für Normalenvektor  $\mathbf{n}$  und Abstand  $d$  vorgegeben werden, die eine Bodenebene oder eine Wandebene klassifizieren.

An dem verwendeten mobilen Roboter (siehe Abbildung 2.1) ist sowohl die Höhe der Kamera als auch deren Nickwinkel einstellbar und damit zur Laufzeit der Wahrnehmungsverfahren bekannt. Ebenen können dementsprechend in Boden- und Wand-Ebenen unterteilt werden. Mit den nicht den Ebenen zugeordneten Punkten sind damit entsprechend der Anforderung alle Punkte den Klassen “Boden,, “Wand,, und “Hindernis“ zugeordnet.

## 3.2 Segmentierung im Bild – 2D

Zahlreiche Bildsegmentierungsverfahren verbessern sich signifikant mit der Genauigkeit der Anfangshypothese. Dazu werden dem Verfahren *Marker* oder Startpunkte<sup>1</sup> vorgegeben, die eine Anfangshypothese für die Bildsegmentierung darstellen und Bildpunkte oder Bildbereiche angeben, deren Zugehörigkeit zu einem der Zielsegmente bereits bekannt ist. Die Bildsegmentierung weitet diese Anfangshypothese entsprechend der Bildinformationen aus.

Vielfach wird in der mobilen Robotik zur Segmentierung der Bodenfläche ein Startpunkt des Bildsegmentierungsverfahrens in der Mitte des unteren Bildrandes einer Frontkamera genutzt. Die dahinter stehende Annahme ist, dass sich unmittelbar vor dem Roboter immer Boden befindet. Ein Startpunkt, der in der Mitte des unteren Bildrandes platziert wird, würde damit ein Bildsegment erzeugen, das den Boden darstellt. Die Annahme, dass sich unmittelbar vor dem Roboter Bodenfläche befindet, muss jedoch nicht immer erfüllt sein und kann in Situationen, in denen der Roboter unmittelbar vor einer Wand oder einem Hindernis steht, zu Fehlsegmentierungen führen. Ein fälschlicherweise als Boden segmentiertes Hindernis könnte zu falschen Bewegungen des Roboters und damit zu Beschädigungen des Roboters oder der Umwelt führen.

In der hier genutzten Konfiguration einer 3D-Kamera und einer monokularen Kamera mit ähnlichen Blickfeldern, kann die bereits erfolgte Segmentierung und Klassifizierung der Messung der 3D-Kamera genutzt werden, um eine sichere Anfangshypothese für die Segmentierungsverfahren zu liefern. Dies ermöglicht auch in Situationen, in denen die klassische Wahl von Startpunkten für die Bildsegmentierung versagt, eine korrekte Wahl von Startpunkten.

---

<sup>1</sup>In der Literatur vornehmlich als „*Seeds*“, engl. etwa “Saatpunkt,, bezeichnet.

Um dies zu erreichen werden die den drei Klassen zugeordneten 3D-Punkte in die monokulare Frontansicht projiziert.

### 3.2.1 Projektion der PMD-Messung

Um die Segmentierung und Klassifizierung in der dreidimensionalen Messung der 3D-Kamera für die Segmentierung im Bild nutzen zu können, werden die Punkte aus der Bildebene der 3D-Kamera in die Bildebene der monokularen Kamera projiziert.

Basis für die Transformation ist die Kalibrierung der Kameras. Diese zerfällt in die intrinsische Kalibrierung der beiden Kameras sowie die extrinsische Kalibrierung zwischen den beiden Kameras. Die epipolare Einschränkung gibt dabei für einen Punkt im Bild der 3D-Kamera eine Linie möglicher Punkte in der Frontansicht. Über die Tiefeninformation, die die 3D-Kamera außerdem für jeden Bildpunkt liefert, kann auch der genaue Punkt in der Kameraansicht ermittelt werden.

Aufgrund der Tatsache, dass die Auflösung der 3D-Kamera im Vergleich zur monokularen Kamera um den Faktor 10 geringer ist, sind jedoch der Genauigkeit der Transformation bereits Grenzen gesetzt. Da zudem die Kalibrierung die intrinsischen Kameraparameter bekannt sind, darüber hinaus die Kameras näherungsweise parallel sind und nur eine geringe Translation (im Verhältnis zur Distanz zu den beobachteten Punkten) aufweisen, ist die Transformation allein im Bildbereich aufgrund der intrinsischen Parameter für die Anwendung ausreichend.

Dabei ist die Transformation ein linearer Zusammenhang, der jeweils in  $x$ - und  $y$ -Richtung über das Verhältnis der Öffnungswinkel der Kameras gegeben ist. Die Öffnungswinkel der Kameras sind proportional zu deren Brennweiten und Chipgrößen. Die Chipgrößen sind bekannt und die Brennweiten  $f_{mono}$  und  $f_{pmd}$  der monokularen beziehungsweise 3D-Kamera wurde über die Kamera-Kalibrierung ermittelt.

Damit sind die Umrechnungsfaktoren  $k$  in  $x$ - und  $y$ -Richtung gegeben durch:

$$k_x = \frac{f_{mono,x}}{f_{pmd,x}}, \quad k_y = \frac{f_{mono,y}}{f_{pmd,y}} \quad (3.4)$$

Der zweidimensionale Bildpunkt  $\mathbf{p}_{mono}$  in der monokularen Frontansicht, der mit einem segmentierten, klassifizierten Punkt  $\mathbf{p}_{pmd}$  des PMD-Sensors korrespondiert, wird damit in ausreichender Näherung bestimmt über

$$\mathbf{p}_{mono} = \begin{bmatrix} k_x \\ k_y \end{bmatrix} \cdot \mathbf{p}_{pmd} \quad (3.5)$$

So lassen sich Punkte über eine einfache lineare Transformation vom PMD-Bild in die monokulare Ansicht transformieren, um dort für die Erzeugung von Anfangshypothesen für Bildsegmentierungsverfahren genutzt zu werden. Abbildung 3.3 zeigt die auf die Weise in die



monokulare Frontansicht projizierten, als Boden klassifizierten Punkte einer Beispielszene.

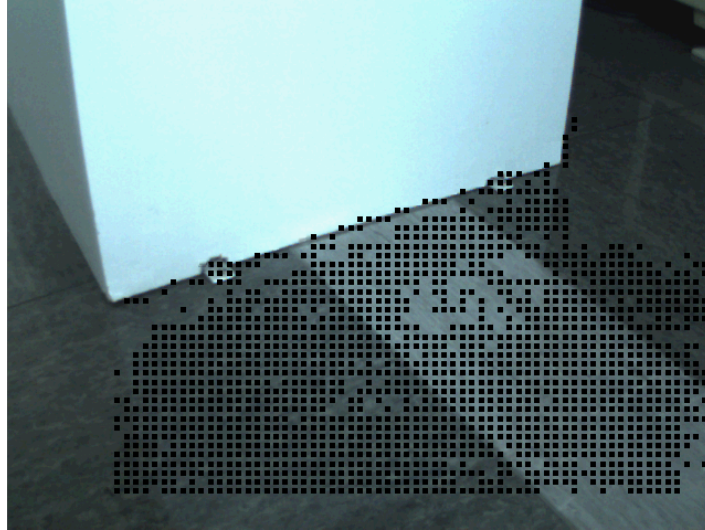


Abbildung 3.3: Mit der 3D-Kamera klassifizierte Punkte transformiert in das monokulare Kamerabild als Startpunkte für die Bildsegmentierung.

### 3.2.2 Segmentierungsverfahren

Um die erläuterten, von der Verarbeitung der 3D-Kamera erhaltenen Anfangshypothesen zur Bildsegmentierung nutzen zu können, sind Segmentierungsverfahren erforderlich, denen eine Startsegmentierung oder Startpunkte vorgegeben werden können. In der Literatur finden sich dafür die Verfahren *Region Growing* und *Marker-based Watershed*.<sup>[11]</sup> Die Segmentierung mit diesen Bildsegmentierungsverfahren wird im Folgenden dargestellt.

#### Region Growing

Ein typisches Verfahren der Bildsegmentierung ist *Region Growing*<sup>2</sup>.<sup>[12]</sup> Dabei wird ausgehend von einem Startpunkt eine Bildregion kontinuierlich ausgeweitet. Alle an die Region (oder anfänglich an den Startpunkt) grenzenden Punkte werden dazu über eine definierte Distanzfunktion daraufhin überprüft, ob sie der Region ähnlich genug sind und hinzugezählt werden. Dazu wird ein Schwellwert der Distanz definiert, unterhalb derer Punkte zu dem Segment hinzugezählt werden. Die Segmentierung ist vollständig, wenn alle an die Region grenzenden Bildpunkte eine höhere Distanz zu der Region aufweisen, als durch den definierten Schwellwert vorgegeben.

Die Distanzfunktion ist in der Regel der euklidische Abstand auf der Intensität der Bildpunkte. Aber auch Distanzen auf einzelnen Farbkanälen, dem Sättigungs- oder Tonwertkanal (bei Farbbildern) werden zur Segmentierung benutzt.<sup>[11]</sup>

---

<sup>2</sup>In der Literatur vielfach auch als *Flood-Fill* bezeichnet.

Ausgangspunkt für *Region Growing* ist immer ein Startpunkt. Typische Anwendung in der Robotik ist die Bodensegmentierung,[13] bei der zum Beispiel ein Punkt in der Mitte des unteren Bildrandes als Startpunkt der Segmentierung angenommen wird. Wie obig dargelegt, ist diese Annahme nicht immer korrekt. Mit den Informationen aus der Segmentierung und Klassifizierung des 3D-Bildes, können jedoch dem Verfahren mehrere (größer 150) Startpunkte übergeben werden, von denen klare Zugehörigkeit zur Zielregion „Boden“ bereits durch die Klassifizierung bekannt ist.

Darüber lassen sich zum einen falsche Startpunkte verhindern und zum anderen mehr Startpunkte erzeugen, um die Ausgangshypothese zu verbessern. Die Anfangshypothese der Segmentierung ist damit um ein vielfaches genauer und sicherer gegenüber Fehlzuordnungen als die klassische Anfangshypothese. In Abbildung 3.4 ist zu sehen, dass dadurch die Segmentierung mit mehreren Startpunkten (rechts) durch *Region Growing* deutlich verbessert wird. Die Segmentierung mit mehreren Startpunkten erkennt gegenüber der Segmentierung mit nur einem Startpunkt (mittlere Abbildung) einen größeren Bildbereich korrekt als Boden. Eine ausführliche Analyse findet sich in Kapitel 5.1.2.

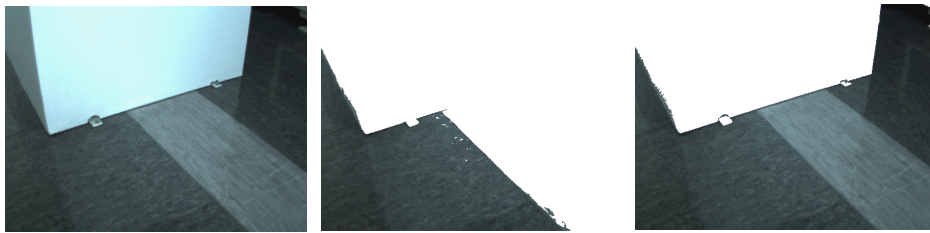


Abbildung 3.4: *Region Growing*: Monokulare Ansicht (Links), Bodensegmentierung mit einem Startpunkt (Mitte) und Bodensegmentierung mit Startpunkten aus der 3D-Klassifizierung.

### Marker-based Watershed

Ein weiteres Verfahren der Bildsegmentierung ist der *Watershed*-Algorithmus<sup>3</sup>. Bei *Watershed* wird das Intensitätsbild als Höhenprofil betrachtet, bei dem der Gradient zwischen Pixeln als *Höhe* einer topologischen Karte betrachtet wird. Heterogene Bildbereiche erhalten hohe Werte, homogene Bildbereiche erhalten niedrige Werte. In diesem Höhenprofil werden mit Markern *Einlaufstellen* definiert, in denen im übertragenen Sinne Wasser einläuft, das sich entsprechend des Höhenprofils ausbreitet.

Die einzelnen Wasserquellen breiten sich demnach quer zu großen Gradienten, also entlang homogener Bildregionen, aus. So wird erreicht, dass sich Anfangshypothesen (die Einlaufstellen) entsprechend des Bildes ausbreiten, bis das komplette Bild segmentiert ist.

*Marker-based Watershed* eignet sich für die vorliegende Aufgabe in besonderem Maße, da gleichzeitig mehrere, konkurrierende Bildsegmente erzeugt werden. Im Gegensatz zu *Region Growing* kann damit nicht nur der Boden segmentiert werden, sondern es werden gleichzeitig

<sup>3</sup> *Watershed*, (engl. „Wasserscheide“)

Bildsegmente für die drei Klassen „Boden“, „Wand“ und „Hindernis“ erzeugt. Damit lassen sich mehr Informationen, die mit der 3D-Kamera gewonnen wurden, übertragen und für die Anfangshypothese der Bildsegmentierung nutzen. Der Algorithmus ist dabei umso robuster je besser und fortgeschrittener die Anfangshypothesen sind.

Um die Marker für den Algorithmus aus den klassifizierten Punkten der 3D-Kameramessung zu erzeugen, werden die Punkte erneut in die monokulare Ansicht projiziert. Um zu verhindern, dass Ausreißer, einzelne durch Rauschen falsch klassifizierte Punkte, die Segmentierung stören, werden dazu die als ein Segment klassifizierten 3D-Punkte im Bildbereich geglättet. So werden Ausreißer eliminiert und zusammenhängende, gleich klassifizierte Punkte zu homogenen Flächen zusammengefasst, die als Marker, also als Anfangshypothese der Segmentierung dienen.

Die in den Bildbereich projizierten Punkte einer Klasse werden dazu gemäß der Gaußverteilung gemittelt:

$$f(u, v) = \frac{1}{2\pi\sigma^2} \cdot e^{-\frac{1}{2}\left(\frac{u^2+v^2}{\sigma^2}\right)} \quad (3.6)$$

Jeder Punkt mit den Pixelkoordinaten  $u$  und  $v$  im Bild, dessen Wert  $f(u, v)$  nach dem Gaußschen Mittelnen einen Grenzwert übersteigt, wird dem Marker zugeordnet. Abbildung 3.5 zeigt, wie in einer Beispielszene daraus Marker für jede der drei Punkte-Klassen entstehen. Die Marker sind durch die Glättung erkennbar kleiner als die tatsächlichen Flächen im Bild, sind dafür aber homogen, zusammenhängend und um Ausreißer bereinigt.



Abbildung 3.5: Watershed-Marker einer Beispielansicht. Von links nach rechts: Monokulare Ansicht, Boden-Marker, Wand-Marker und Hindernis-Marker.

In einer Szene, die Boden, Wände und Hindernisse enthält, werden dadurch drei Marker erzeugt. Diese Marker werden anschließend zu einem Gesamtmarker kombiniert, der einzelne Segmente als Boden, Wand oder Hindernis markiert. Dieser Marker dient als Anfangshypothese für die Segmentierung mittels *Marker-based Watershed*.

Die aus den Markern resultierende Segmentierung segmentiert das Bild vollständig und besitzt soviele Segmente wie übergebene Marker. Im Fall dieser Anwendung also maximal drei Segmente, wenn sich im Bild sowohl Boden als auch Wände und Hindernisse befinden. Die einzelnen Segmente lassen sich über die Marker, aus denen sie gewachsen sind, den Klassen „Boden“, „Wand“ oder „Hindernis“ zuweisen. Abbildung 3.6 zeigt das Ergebnis einer solchen

Segmentierung in einer Beispielsicht.

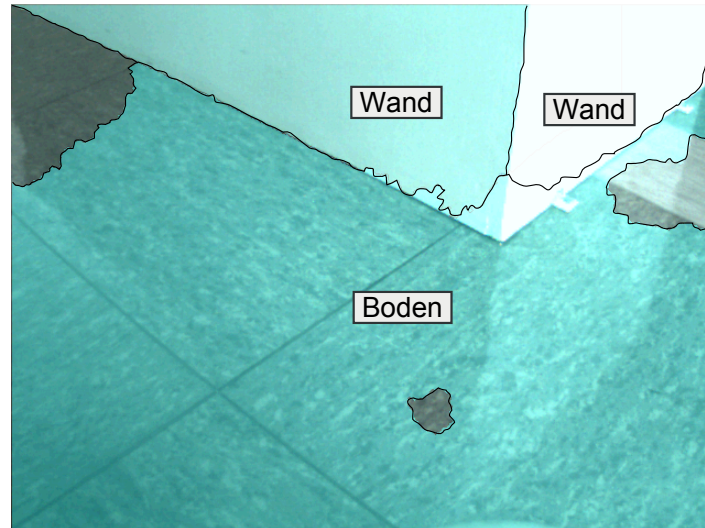


Abbildung 3.6: Monokulare Ansicht segmentiert mit *Marker-Based Watershed*

### 3.2.3 Labeln der Bilder

Sind über die Bildsegmentierungs-Verfahren die Bildsegmente gefunden, können sie anhand der Informationen der 3D-Kamera klassifiziert werden. Ein Bildsegment wird mit derjenigen Klasse („Boden“, „Wand“ oder „Hindernis“) gelabelt, die die meisten der im Segment enthaltenen klassifizierten 3D-Punkte aufweisen. Abbildung 3.6 zeigt die mit den Labeln beschrifteten Segmente der monokularen Frontansicht.

## 3.3 Automatisches Bewerten der Segmentierung

Zur Segmentierung von Bildern sind über die zwei genannten hinausgehend mehrere Verfahren bekannt, die unterschiedliche Vorteile besitzen und in verschiedenen Situationen unterschiedlich gute Ergebnisse erzielen.[\[11\]](#)

Für die lokale Umgebungsrepräsentation wird im Anschluss an die Segmentierung der Frontansicht eine Segmentierung der omnidirektionalen Ansicht durchgeführt. Diese beruht auf den gleichen Verfahren wie die Segmentierung der Frontansicht. Sie besitzt jedoch im Vergleich zur Bildgröße wesentlich kleinere und unpräzisere Anfangshypothesen, da die Überschneidung der Sichtfelder der 3D-Kamera und der omnidirektionalen Kamera nur einen Bruchteil der omnidirektionalen Ansicht ausmachen (siehe Abbildung 3.1).

Um trotz der schwachen Anfangshypothese eine gute Segmentierung der omnidirektionalen Ansicht zu gewährleisten, kann anhand der Segmentierung der monokularen Ansicht eine situationsabhängige Bewertung der einzelnen Segmentierungsverfahren vorgenommen wer-

den. Die dahinter stehende Annahme ist, dass sich die von der monokularen Kamera und von der omnidirektionalen Kamera beobachtete Umgebung wenig unterscheidet. Ein in der monokularen Ansicht erfolgreiches Bildsegmentierungsverfahren erzielt demnach auch in der omnidirektionalen Ansicht gute Ergebnisse.

Die Bewertung der Bildsegmentierung in der monokularen Frontansicht beruht auf der Kenntnis, dass mit der Segmentierung und Klassifizierung der Aufnahme der 3D-Kamera für einzelne Bildbereiche bereits eine Referenz bekannt ist. Die Segmentierungsverfahren werden darauf überprüft, inwieweit die resultierende Segmentierung mit dieser Referenz übereinstimmt.

Dafür wurde ein Bewertungsschema für Segmentierungsverfahren anhand der 3D-Punkte entworfen, das sowohl eine Segmentierung zu großer und damit gegebenenfalls falscher Flächen („falsch positiv“), als auch die Segmentierung zu kleiner Flächen („falsch negativ“) bestraft.

Bildsegmentierungsverfahren, die wie *Region Growing* oder *Marker-Based Watershed* auf eine Anfangshypothese angewiesen sind, erhalten dazu nur einen Teil (zum Beispiel ein Drittel) der klassifizierten 3D-Punkte als Startpunkte oder Marker. Anhand der übrigen, nicht als Anfangshypothese genutzten, Punkte wird die Segmentierung überprüft. Dabei wird gezählt

1. wie viele der als dem Segment zugehörigen PMD-Punkte fälschlicherweise nicht durch die Segmentierung erreicht wurden und
2. wie viele der als anderes Segment klassifizierten PMD-Punkte fälschlicherweise durch die Segmentierung erreicht wurden.

Nicht erreichte, aber als dem Segment zugehörig bekannte Punkte (siehe 1.) bedeuten eine zu schwache Segmentierung und werden einfach bestraft. Von der Segmentierung erreichte Punkte, die als einem anderen Segment zugehörig bekannt sind, (siehe 2.) werden doppelt bestraft. Die doppelte Gewichtung des zweiten Falls (falsch Positive) hat seine Ursache in der Auswirkung auf die Anwendung. Eine Segmentierung, die zu große Flächen segmentiert, kann im Zweifelsfall zu große Flächen fälschlicherweise als Boden und damit als für den Roboter befahrbar klassifizieren. Dies muss unbedingt vermieden werden, um Beschädigungen des Roboters und der Umwelt zu vermeiden.

Sei  $n_{bottom}$  die Anzahl aller als Boden klassifizierten 3D-Punkte,  $n_{other}$  die Anzahl aller als Wand oder Hindernis klassifizierten 3D-Punkte,  $n_{fp}$  die Anzahl der falsch Positiven der Bodensegmentierung und  $n_{fn}$  die Anzahl der falsch Negativen der Bodensegmentierung. Dann gibt

$$b_{fn} = \frac{n_{fn}}{n_{bottom}} \quad (3.7)$$

die Quote der falsch Negativen bezogen auf die Anzahl aller als Boden klassifizierten 3D-Punkte an. Das Verhältnis

$$b_{fp} = \frac{n_{fp}}{n_{other}} \quad (3.8)$$

gibt die Quote der falsch Positiven bezogen auf die Anzahl aller als Wand oder Hindernis klassifizierten 3D-Punkte an.

Falsch Positive werden doppelt gewichtet, dementsprechend erfolgt die Bewertung eines Segmentierungsverfahren anhand der folgenden Formel:

$$b = 1 - (2 * b_{fp} + b_{fn})/3 \quad (3.9)$$

Die Bewertung  $b$  liegt damit in dem Wertebereich  $[0, 1]$ .  $b = 1$  bedeutet eine genaue Segmentierung ohne falsch Positive und falsch Negative,  $b = 0$  bedeutet eine gänzliche falsche Segmentierung ohne korrekte Zuordnungen. Tabelle 3.1 zeigt die Bewertung der Segmentierungsverfahren *Region Growing* und *Marker-Based Watershed* anhand Formel 3.9 in einer Beispielsansicht.

Verfahren	$n_{fn}$	$n_{fp}$	$b_{fn}$	$b_{fp}$	$b$
<i>Region Growing</i>	168	191	0,12	0,19	<b>0,84</b>
<i>Marker-Based Watershed</i>	181	31	0,15	0,07	<b>0,90</b>

Tabelle 3.1: Beispiel der Bewertung zweier Segmentierungsverfahren

*Marker-Based Watershed* erhält in diesem Fall eine bessere Bewertung als *Region Growing*, da mehr Punkte korrekt segmentiert und weniger Punkte falsch segmentiert werden. Zur Segmentierung der omnidirektionalen Ansicht erhält *Marker-Based Watershed* daher in dieser Beispielszene den Vorzug.

Die automatische Bewertung erlaubt die Auswahl der besten Bildsegmentierungsverfahrens in einer konkreten Situation auf Basis der Segmentierung und Klassifizierung der 3D-Informationen. Zu beachten ist hierbei jedoch, dass dieses Bewertungsschema nur in Situationen eine adäquate Bewertung zulässt, in denen mindestens zwei der drei möglichen Bildbereiche (Boden, Wand, Hindernis) in der Frontansicht sichtbar sind. Nur in diesem Fall können falsch Positive bei der Bewertung erkannt und berücksichtigt werden. Ist in der Frontansicht zum Beispiel nur Boden sichtbar, erhalten Segmentierungsverfahren, die bei Wänden oder Hindernissen im Bild falsch Positive produzieren würden, trotzdem eine gute Bewertung. Wird die automatische Bewertung der Bildsegmentierungsverfahren zur Laufzeit des Roboterverhaltens eingesetzt, muss also die Bewertung der jeweils letzten Szene verwendet werden, in der entweder eine Wand oder ein Hindernis in der Frontansicht sichtbar waren.

### 3.4 Omnidirektionale Ansicht

Zur Erzeugung einer lokalen Umgebungsrepräsentation wird eine Segmentierung in der omnidirektionalen Ansicht analog zur monokularen Frontansicht durchgeführt. Die segmentierte omnidirektionale Ansicht erlaubt eine Repräsentation der lokalen Umgebung des Roboters und bietet damit eine Basis für die Navigation, insbesondere für das Korridorfolgeverhalten, da dafür die Position des Roboters in seiner Umgebung ausschlaggebend ist.

### 3.4.1 Transformation

Analog zur Transformation der 3D-Punkte in die monokulare Frontansicht, werden die klassifizierten 3D-Punkte in die omnidirektionale Ansicht übertragen. Dazu wird das omnidirektionale Bild zunächst *aufgefaltet*. Die Auffaltung erfolgt auf Basis einfacher trigonometrischer Umrechnung, eine Kalibrierung der omnidirektionalen Kamera ist nicht notwendig. Die Umrechnung ist in Abbildung 3.7 dargestellt.

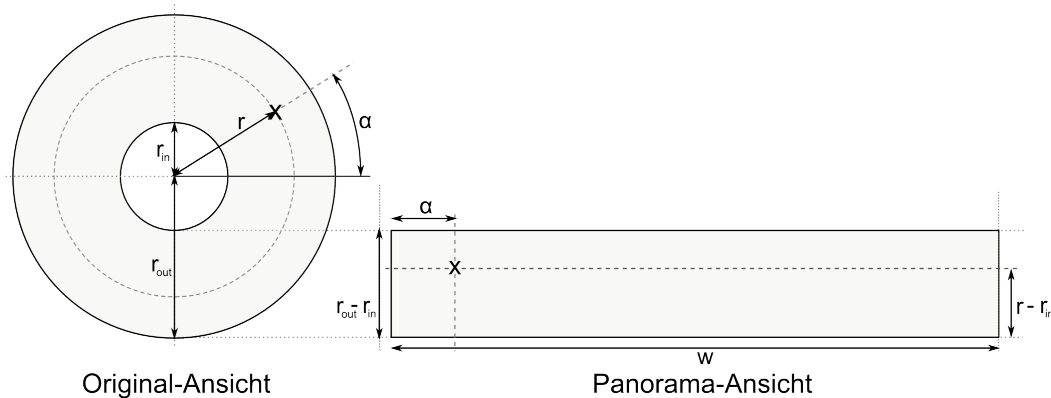


Abbildung 3.7: Beziehung zwischen abgerollter und originaler omnidirektionaler Ansicht.

Die Umrechnung der verzerrten, omnidirektionalen Ansicht in die Panoramaansicht erfolgt aufgrund der Größe des omnidirektionalen Bildes, sowie aufgrund des inneren und äußeren Radius des Bildrings der omnidirektionalen Ansicht. Durch die Kenntnis über die Geometrie dieses Ringes kann dieser in eine Panoramaansicht *abgerollt* werden. Abbildung 3.7 zeigt die zugrunde liegende Beziehung zwischen der aufgenommenen Ansicht, dem omnidirektionalen Ring, und der abgerollten Panoramaansicht. Ein Punkt mit den kartesischen Pixelkoordinaten  $(u, v)$  in der Panoramaansicht korrespondiert mit einem Punkt der Polarkoordinaten  $(r_{out} - v, \frac{u}{w} * 2\pi)$  in der originalen Ansicht.  $w$  ist die Breite des Panoramabildes,  $r_{out}$  der äußere Radius der Original-Ansicht.

Das Auffalten der Ansicht erleichtert zum einen die Korrespondenz zwischen Frontansicht und omnidirektionaler Ansicht, weil die radiale Verzerrung eliminiert wird, zum anderen erlaubt das entstandene Panoramabild die Anwendung gebräuchlicher Segmentierungsverfahren, die für Panoramabilder ausgelegt sind.[\[14\]](#)

Zu beachten ist bei der Auffaltung, dass sie nicht fehlerfrei geschieht, da gleiche Raumbereiche in den beiden Ansichten auf Bildbereiche unterschiedlicher Auflösung abgebildet werden. Der obere und untere Bildrand haben in der Panorama-Ansicht dieselbe horizontale Auflösung, der korrespondierende innere und äußere Radius in der originalen Ansicht weisen deutlich unterschiedliche Auflösungen auf.

Da die Panorama-Ansicht vornehmlich für die Segmentierung des Bodens eingesetzt wird, wird die horizontale Auflösung der Panorama-Ansicht so gewählt, dass im unteren Bildbereich, also in Umgebung des Bodens, die Verzerrung möglichst gering ist. Dies wird erreicht, indem die

horizontale Auflösung der Panorama-Ansicht der Auflösung des inneren Radius in der originalen Ansicht entspricht. So ist die Verzerrung in der untersten Zeile der Panorama-Ansicht eliminiert, da sie die gleiche Auflösung wie der korrespondierende innere Radius der originalen Ansicht aufweist. Die Verzerrung nimmt zu den oberen Bildbereichen der Panorama-Ansicht zu.

### 3.4.2 Segmentierung

Ist die omnidirektionale Ansicht zu einem Panoramabild aufgefaltet, kann die Korrespondenz zwischen dem 3D-Kamerabild und dem Bild der monokularen Kamera auf der einen Seite und der aufgefalteten omnidirektionalen Ansicht auf der anderen Seite analog zu Kapitel 3.2 einfach über einen horizontalen und vertikalen Offset und eine horizontale und vertikale Streckung oder Stauchung reduziert werden.

Der horizontale Offset zwischen Panoramabild und monokularer Ansicht beträgt die halbe Breite des Panoramabildes abzüglich der halben Breite der monokularen Frontansicht. Der vertikale Offset hängt vom Nickwinkel der monokularen Kamera ab. Horizontale Streckung oder Stauchung hängen von der gewählten Auflösung des Panoramabildes und dessen Verhältnis zur Auflösung der Frontansicht ab.

Zur Segmentierung des Bildes wird die vormals in der monokularen Ansicht durchgeführte Bewertung der Segmentierungsverfahren genutzt. So wird die omnidirektionale Ansicht mit den nach Kapitel 3.3 best-bewerteten Segmentierungsverfahren durchgeführt. Die zugrunde liegende Annahme dafür ist, dass die Erscheinung der Umgebung in der omnidirektionalen Ansicht und die in der monokularen Ansicht hinreichend ähnlich sind und zudem gut bewertete Segmentierungsverfahren der monokularen Frontansicht aufgrund der Auffaltung der omnidirektionalen Ansicht auch in der omnidirektionalen Ansicht gute Ergebnisse liefern.

Abbildung 3.8 zeigt das Ergebnis einer so segmentierten, omnidirektionalen Panorama-Ansicht. Zu sehen ist unten im Bild die segmentierte Bodenfläche, die damit zur Navigation genutzt werden kann.



Abbildung 3.8: Segmentierte omnidirektionale Ansicht der lokalen Umgebung. Das Bodensegment ist optisch hervorgehoben und dient als Grundlage zum Beispiel für die Korridorfolge.



# Kapitel 4

## Roboterverhalten

Im Rahmen dieser Arbeit wird ein robustes Roboterverhalten basierend auf bildgebender Sensorik trainiert. Grundlage für die Wahrnehmung ist die dargestellte robuste Segmentierung der monokularen Frontansicht und der omnidirektionalen Umgebungsansicht aus Kapitel 3.

Gelernt werden soll ein sogenanntes reaktives Verhalten, ein Verhalten, das in Abgrenzung zu planenden Verhalten ausschließlich auf den aktuellen Wahrnehmung beruht und keine inneren Zustände besitzt. Das reaktive Verhalten wird durch maschinelles Lernen aufgrund aufgezeichneter Trainingsdaten erzeugt. Beispielhaft wird im Folgenden ein Verfahren zur Kollisionsvermeidung und Korridorfolge gelernt. Kollisionsvermeidung und Korridorfolge werden dabei nicht separat, sondern in Kombination gelernt.

Kollisionsvermeidung ist eine typische Disziplin der mobilen Robotik, die dem Roboter Fortbewegung ohne Kollision und damit ohne Beschädigung des Roboters oder der Umwelt ermöglicht. Dazu müssen Hindernisse in Abgrenzung zur frei befahrbaren Fläche erkannt werden und entsprechende Bewegungsanweisungen für den Roboter erzeugt werden, die ihn kollisionsfrei über befahrbare Fläche bewegen.

Korridorfolge ist eine weitere typische Disziplin der mobilen Robotik in Innenraum-Umgebungen, zum Beispiel Büroumgebungen. Dabei geht es darum, dass der Roboter beim Durchfahren eines Flurs diesen erkennt und sich möglichst mittig im Flur bewegt. Analog zur Korridorfolge in Innenraum-Szenarien verhalten sich Pfadfolge-Verhalten in Außen-Umgebungen.

Wie in Kapitel 2.3 dargestellt, wird das Verhalten basierend auf einer Trainingsdatenbasis gelernt. Die Trainingsdatenbank ist eine Sammlung von Instanzen, die jeweils aus einem Merkmalsvektor und der zu lernenden Ausgangsgröße bestehen. Sind ausreichend viele Trainingsdaten vorhanden, wird daraus über maschinelles Lernen das Verhalten generalisiert.

In Kapitel 2.3.2 wurde erwähnt, welche Nachteile und Probleme bei instanzbasiertem Lernen beachtet werden müssen. Um dies zu tun muss im Folgenden

1. eine Glättung zur Reduktion des Rauschens geschehen,
2. auf die Wertebereiche des Merkmals geachtet werden,

3. bei der Auswahl der Merkmale eine Beschränkung auf ausschließlich relevante Merkmale stattfinden und
4. der hohe Rechenaufwand zur Abfragezeit berücksichtigt werden.

## 4.1 Ausgangsgröße

Bei mobilen Robotern wird typischerweise die Kombination aus Vorwärtsgeschwindigkeit und Rotationsgeschwindigkeit des Roboters als Zielgröße eines Verhaltens definiert. Dies sind auch die Größen, die dem *Pioneer 3-DX* als Motorkommandos übermittelt werden.

Tatsächliche Zielgröße eines Kollisionsvermeidungs- und Korridorfolgeverhaltens ist jedoch weniger die gefahrene Geschwindigkeit als die gefahrene Trajektorie, die Kollision mit Hindernissen vermeidet und im Korridor mittig verläuft. Dementsprechend kann als zu lernende Ausgangsgröße des maschinellen Lernens eine Größe gewählt werden, die direkt die Trajektorie repräsentiert.

Eine Trajektorie lässt sich vollständig durch die Krümmung und den Normalenvektor in jedem ihrer Punkte beschreiben. Da sich der mobile Roboter nur in einer Ebene bewegt, bleibt der Normalenvektor konstant. In diesem Fall kann die Trajektorie in der Ebene durch die Krümmung in jedem ihrer Punkte beschrieben werden. Die Krümmung einer gefahrenen Trajektorie in einem Punkt ist definiert als Richtungsänderung pro Längeneinheit, siehe Abbildung 4.1.

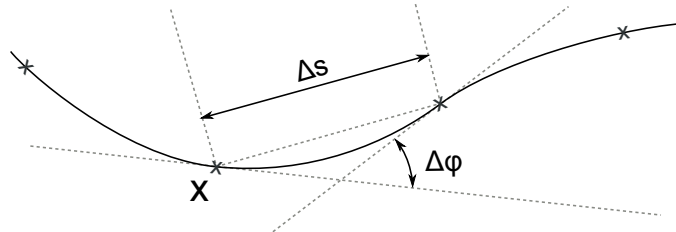


Abbildung 4.1: Krümmung einer Trajektorie in einem Punkt  $x$ , gegeben durch die Wegdifferenz  $\Delta s$  und die Differenz der Orientierung  $\Delta\varphi$ .

Mit  $v = \frac{\Delta s}{\Delta t}$  als Änderung des Weges in einem Zeitschritt und  $\omega = \frac{\Delta\varphi}{\Delta t}$  als Änderung der Orientierung in einem Zeitschritt, sowie einer Vorwärtsgeschwindigkeit  $v$  über Null, ist die Bahnkrümmung gegeben durch

$$\kappa = \frac{\Delta\varphi}{\Delta s} = \frac{\omega}{v} \quad (4.1)$$

Die Ausgangsgröße  $\kappa$  des maschinellen Lernens kann damit zu jedem Zeitpunkt direkt aus den Motorkommandos  $v$  und  $\omega$  des Roboters ermittelt werden. Die Rotationsgeschwindigkeit  $\omega$  des Roboters ist gegeben in der Einheit  $\frac{\circ}{s}$  (Grad pro Sekunde), die Vorwärtsgeschwindigkeit  $v$  ist gegeben in der Einheit  $\frac{m}{s}$ . Die Ausgangsgröße  $\kappa$  hat dementsprechend die Einheit  $\frac{\circ}{m}$ .

Ein Vorteil der Krümmung  $\kappa$  und damit der Trajektorie als Ausgangsgröße des Lernens ist, dass nunmehr die Geschwindigkeit des Roboters in Form einer Grundgeschwindigkeit von außen zur Laufzeit des Verhaltens vorgegeben werden kann und nicht von der gefahrenen Geschwindigkeit beim Aufzeichnen der Trainingsdaten abhängig ist. Wird die Rotationsgeschwindigkeit  $\omega$  als Basisgeschwindigkeit vorgegeben, kann über die Formel 4.1 die korrespondierende Vorwärtsgeschwindigkeit  $v$  zum Fahren der Zieltrajektorie zur Laufzeit errechnet werden. Analog dazu kann die Rotationsgeschwindigkeit bei einer vorgegebenen Vorwärtsgeschwindigkeit ermittelt werden.

## 4.2 Auswahl geometrischer Merkmale

Grundlage für den Trainingsvektor und damit für den maschinellen Lernprozess ist neben der Ausgangsgröße der Merkmalsvektor. Er bestimmt die Dimension des Instanzraumes und repräsentiert die Umwelt für den Roboter in einer reduzierten, für die Aufgabe relevanten Art und Weise.

Die gewählten Merkmale spannen einen Instanzraum auf. In diesem Instanzraum werden beim instanzbasierten Lernen die einzelnen Instanzen entsprechend ihren Ausprägungen der Merkmale abgelegt. Die Auswahl der Merkmale bestimmt damit essentiell den Lernprozess. Die Distanzfunktion als Grundlage für das Finden der nächsten Nachbarn und damit als Grundlage für die Klassifikation oder Regression ist in dem Instanzraum, also auf dem Merkmalsvektor, definiert.

Da die Distanzfunktion eine Funktion der Distanzen in den einzelnen Dimensionen des Instanzraumes ist, ist der Wertebereich einzelner Merkmale relevant für die Klassifikation oder Regression. Ein Merkmal mit einem größeren Wertebereich beeinflusst die Distanzfunktion stärker als Merkmale mit kleinem Wertebereich.[8] Alle in dieser Arbeit verwendeten Merkmale werden daher auf den einheitlichen Wertebereich  $[0, 1]$  normiert.

### 4.2.1 Manuelle Auswahl

Die Auswahl der Merkmale muss ein korrektes Verhalten ermöglichen, die Problemklasse also angemessen repräsentieren. Eine Möglichkeit der Auswahl geeigneter Merkmale ist die manuelle Auswahl aufgrund von Expertenwissen.

Als Basis dienen dafür Erfahrungen mit manuell implementierten Roboterverhalten. Zu Beginn der Arbeit wurde die in Kapitel 3 dargestellte Wahrnehmung mit programmierten, nicht gelernten Roboterverhalten erprobt. Dazu wurden händisch Merkmale extrahiert, die in programmierten Korridorfolge- und Kollisionsvermeidungsverhalten ihre generelle Tauglichkeit gezeigt haben, also eine angemessene Repräsentation der Umwelt darstellen.

Diese Merkmale wurden daher als Merkmalsvektor für das instanzbasierte Lernen eines kombinierten Korridorfolge- und Kollisionsvermeidungsverhaltens verwendet. Die manuelle Auswahl der Merkmale wird im Folgenden dargestellt. Bei den programmierten Verhalten sind Kollisionsvermeidung und Korridorfolge in Form zweier getrennter Verhalten implementiert. Der

Merkmalsvektor für das Lernen des kombinierten Korridorfolge- und Kollisionsvermeidungsverhaltens ist die Kombination der für die zwei Verhalten ausgewählten Merkmale.

### Merkmale für die Kollisionsvermeidung

Für das Kollisionsvermeidungsverhalten ist der Bereich des Bodens in Fahrtrichtung des Roboters relevant. Dort befindliche Hindernisse muss der Roboter bei der weiteren Fahrt berücksichtigen, um eine Kollision zu vermeiden. Die Merkmale für ein Kollisionsvermeidungsverhalten bieten demnach eine geeignete Repräsentation der Bodenfläche unmittelbar vor dem Roboter in Fahrtrichtung. Die Merkmale gründen sich entsprechend auf der Bodensegmentierung der monokularen Frontansicht gemäß Kapitel 3.2.

Das Bild der monokularen Kamera wird entsprechend segmentiert und in sieben Spalten eingeteilt. In jeder Spalte des Bildes wird die Höhe des Bodens im Bild  $h_i$  bestimmt. Die folgende Abbildung 4.2 zeigt die Ausprägung der Merkmale anhand einer Beispielinstantz.

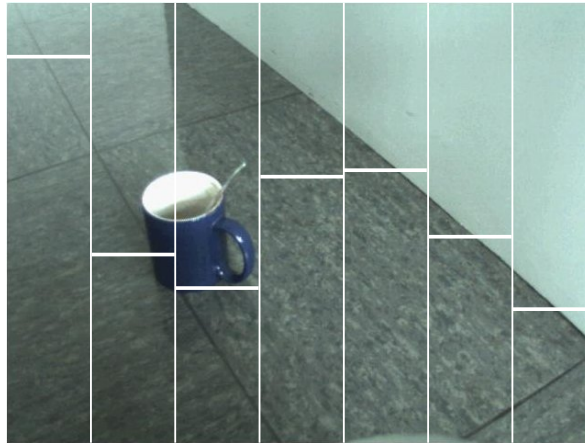


Abbildung 4.2: Zu sehen sind die sieben Spalten und jeweils die Höhe des Bodens. In Richtung der Hindernisse ergeben sich niedrige Werte, in Richtung freien Bodens hohe Werte.

$h_1$	$h_2$	$h_3$	$h_4$	$h_5$	$h_6$	$h_7$
0,87	0,43	0,35	0,61	0,62	0,57	0,31

Tabelle 4.1: Ausprägungen der sieben Merkmale am Beispiel.

Tabelle 4.1 zeigt die daraus extrahierten Merkmalswerte, normiert auf den Wertebereich  $[0, 1]$  (1 stellt die Bildhöhe dar). Links im Bild befindet sich ein Hindernis, rechts im Bild eine Wand. Die höchsten Merkmalswerte befinden sich dementsprechend links vom Hindernis und zwischen Wand und Hindernis.

### Merkmale für die Korridorfolge

Für das Korridorverhalten muss eine entsprechende Repräsentation des Korridors gefunden werden. Hierfür ist nicht nur die Bodenfläche direkt vor dem Roboter relevant, sondern die gesamte, den Roboter umgebene Bodenfläche. Dementsprechend werden die Merkmale für die Korridorfolge aus der omnidirektionalen Umgebungsansicht extrahiert.

Der Roboter befindet sich mittig in einem Korridor positioniert, wenn sich zu seiner rechten und linken Seite gleichviel Bodenfläche befindet. Er ist darüber hinaus in Richtung des Korridors ausgerichtet, wenn sich entlang seiner Fahrtrichtung mehr Bodenfläche befindet als quer zur Fahrtrichtung.

Entsprechend dem Verhältnis von Bodenfläche links und rechts des Roboters kann die notwendige Trajektorie zum Erreichen der Flurmitte ermittelt werden. Dementsprechend wird dieses Verhältnis als ein Merkmal gewählt, sowie als weiteres Merkmal das Verhältnis von Bodenfläche in Fahrtrichtung zur Bodenfläche quer zur Fahrtrichtung.

Zur Berechnung der Merkmalswerte einer Instanz wird die omnidirektionale Ansicht horizontal in acht Segmente unterteilt. In jedem Segment wird analog zum Vorgehen in der Frontansicht die Höhe  $h_{(o,i)}$  des Bodens im Bild ermittelt.<sup>1</sup> Das Verhältnis der Bodenfläche links des Roboters zur Bodenfläche rechts des Roboters ist das Verhältnis der Werte in den Spalten 1 – 4 (zusammengefasst zur Bodenfläche links im Bild  $a_{(o,left)}$ ) zu den Werten in den Spalten 5 – 8 (Bodenfläche rechts im Bild  $a_{(o,right)}$ ). Das Verhältnis der Bodenfläche in Fahrtrichtung des Roboters  $a_{(o,long)}$  zur Bodenfläche quer zur Fahrtrichtung des Roboters  $a_{(o,lat)}$  ist das Verhältnis der Werte in den Spalten 1, 4, 5 und 8 zu den Werten in den Spalten 2, 3, 6 und 7.

Das Verhältnis von  $a_{(o,long)}$  zu  $a_{(o,lat)}$  gibt ein Maß, inwieweit der Roboter in Richtung des Korridors ausgerichtet ist. Als Merkmal  $a_{(o,corr)}$  normiert auf den Bereich  $[0, 1]$  ergibt das:

$$a_{(o,corr)} = \frac{a_{(o,long)}}{a_{(o,long)} + a_{(o,lat)}} \quad (4.2)$$

Das Verhältnis von  $a_{(o,left)}$  zu  $a_{(o,right)}$  gibt an, ob der Roboter sich rechts oder links der Korridormitte befindet:

$$a_{(o,tr)} = \frac{a_{(o,left)}}{a_{(o,left)} + a_{(o,right)}} \quad (4.3)$$

Abbildung 4.3 zeigt eine Beispielinstantz, Tabelle 4.2 zeigt die entsprechende Ausprägung der Merkmale in dieser Beispielinstantz.

Zu sehen ist, dass der Roboter sich in Fahrtrichtung des Korridors befindet – erkennbar daran, dass das erste Merkmal einen Wert größer 0,5 aufweist (mehr Bodenfläche in Fahrtrichtung

<sup>1</sup>Der Index  $o$  kennzeichnet Merkmale der omnidirektionalen Ansicht in Abgrenzung zu den aus der Frontansicht extrahierten Merkmalen.



Abbildung 4.3: Bodensegmentierung in der omnidirektionalen Ansicht. Aus der Segmentierung kann die Verteilung des Bodens entsprechend der Merkmale ermittelt werden.

$a_{(o,corr)}$	$a_{(o,lr)}$
0,60	0,37

Tabelle 4.2: Ausprägungen der zwei Merkmale am Beispiel.

des Roboters als quer zur Fahrtrichtung). Der Roboter befindet sich außerdem etwas linksseitig im Flur – erkennbar daran, dass das zweite Merkmal einen Wert kleiner 0,5 aufweist (mehr Bodenfläche rechts- als linksseits des Roboters).

### Merkmalsvektor

Der manuell ausgewählte Merkmalsvektor für ein kombiniertes Kollisionsvermeidungs- und Korridorfolgeverhalten gründend auf Erfahrungen mit den programmierten Einzelverhalten, ist somit der 9-dimensionale Vektor mit den Merkmalen:  $a_{(o,corr)}$ ,  $a_{(o,lr)}$ ,  $h_1$ ,  $h_2$ ,  $h_3$ ,  $h_4$ ,  $h_5$ ,  $h_6$ ,  $h_7$ .

#### 4.2.2 Automatische Auswahl

Die manuelle Auswahl geeigneter geometrischer Merkmale ist ein heuristischer Prozess, der zu Merkmalen führen kann, die nur scheinbar relevant sind oder redundant zu anderen Merkmalen. Instanzbasiertes Lernen ist jedoch empfindlich gegenüber irrelevanten Merkmalen, eine Vermeidung solcher Merkmale ist daher essentiell für gute Ergebnisse. Die Wahl falscher (irrelevanter, redundanter) Merkmale muss demnach durch einen geeigneten Auswahlprozess verhindert werden.

Eine Lösung, den Merkmalsvektor nur aus nachweisbar relevanten Merkmalen aufzubauen, stellen existierende Verfahren zur automatischen Merkmalsauswahl dar. Eine Übersicht zu Verfahren der automatischen Merkmalsauswahl findet sich bei Molina et al.[15]. Das gebräuchlichste Verfahren zur automatischen Merkmalsauswahl ist die Auswahl mittels Vorwärtsverkettung (*Forward Sequential Selection*, FSS).[16] Das Verfahren beruht auf dem sukzessiven Hinzufügen des nächst-relevanten Merkmals zu einem anfangs leeren Merkmalsvektor. Dazu wird jeweils ein Merkmal dem Merkmalsvektor temporär zugefügt und der resultierende Gesamtfehler des maschinellen Lernens ermittelt. Das Merkmal, welches die deutlichste Reduktion des Fehlers bewirkt, wird als Merkmal dauerhaft dem Merkmalsvektor zugefügt.

Als Abbruchkriterium kommen verschiedene Optionen in Betracht. Die Merkmalsauswahl

kann zum Beispiel beendet werden, wenn eine vorgegebene Anzahl von Merkmalen erreicht ist. Dies kann nützlich sein, um die Dimension des Instanzraumes zu begrenzen und damit auch den Rechenaufwand und die Zeit einer Abfrage. Weiterhin kann als Abbruchkriterium der Merkmalsauswahl definiert werden, wenn ein vorgegebener Wert des Lernfehlers unterschritten ist, das Lernverfahren also vorgegebene Gütekriterien erfüllt. Als dritte Möglichkeit kann die Merkmalsauswahl abgebrochen werden, sobald durch Hinzufügen weiterer Merkmale keine signifikante Reduktion oder sogar eine Zunahme des Fehlers auftritt. Zu einer Zunahme des Fehlers bei Hinzufügen weiterer Merkmale kann es zum Beispiel durch die sogenannte Überanpassung kommen.

Zu beachten ist, dass FSS nicht zwingend zum optimalen Merkmalsvektor findet, da nicht sämtliche möglichen Kombinationen von Merkmalen als Merkmalsvektor getestet werden. So wird mit der Auswahl des ersten, relevantesten Merkmals die Suche bereits auf sämtliche Merkmalsvektoren beschränkt, die dieses Merkmal enthalten.

Der Forderung, den Merkmalsvektor auf relevante Merkmale zu beschränken, um gute Lernergebnisse zu erzielen, ist dennoch entsprochen, da durch die Entwicklung des Lernfehlers der Punkt beobachtet werden kann, an dem sich die Lernergebnisse durch Hinzufügen des nächst-relevanten Merkmals verschlechtern.

### **Merkmalskandidaten**

Für die automatische Auswahl der Merkmale zur Erfüllung der vorliegenden Aufgabe – ein robustes Kollisionsvermeidungs- und Korridorfolgeverhalten zu implementieren – wurden mehrere Merkmale auf Basis der Segmentierung der monokularen Frontansicht und der omnidirektionalen Ansicht erstellt. Auf dieser Menge an Merkmalskandidaten wird die automatische Merkmalsauswahl mittels FSS durchgeführt.

Zu den zur Verfügung stehenden Merkmalen gehören die in Kapitel 4.2.1 dargestellten, manuell ausgewählten Merkmale, sowie weitere Merkmale, die die Struktur der Umgebung darstellen. Die folgende Liste zeigt beispielhaft einige der zur automatischen Merkmalsauswahl zur Verfügung gestellten Merkmale:

1. Höhe des Bodens in der Frontansicht in zwölf Spalten
2. Höhe des Bodens in der omnidirektionalen Ansicht in acht Spalten
3. Verhältnis des Bodens links/rechts in der Frontansicht
4. Verhältnis des Bodens links/rechts in der omnidirektionalen Ansicht
5. Relative Menge Boden/Wand/Hindernis in der Frontansicht

Insgesamt wurden 48 verschiedene Merkmale aus den mittels Bildverarbeitung segmentierten Ansichten extrahiert. Eine vollständige Liste findet sich in Kapitel A.

### Merkmalsvektor

Das instanzbasierte Lernen findet nicht im 48-dimensionalen Instanzraum statt, sondern aus den 48 zur Verfügung stehenden Merkmalskandidaten werden automatisch die relevantesten Merkmale extrahiert. Dazu werden aus Trainingsfahrten in unterschiedlichen Umgebungen die vollen Merkmalsvektoren mit 48 Merkmalen generiert und anschließend offline mit den Verfahren der automatischen Merkmalsauswahl die relevantesten Merkmale extrahiert.

Abbildung 4.4 zeigt den Verlauf des durchschnittlichen Lernfehlers  $E$  über die Anzahl der per FSS dem Merkmalsvektor hinzugefügten Merkmale. Bei jedem Merkmalsvektor wird  $E$  als arithmetisches Mittel des Fehlers über alle Testinstanzen bestimmt.

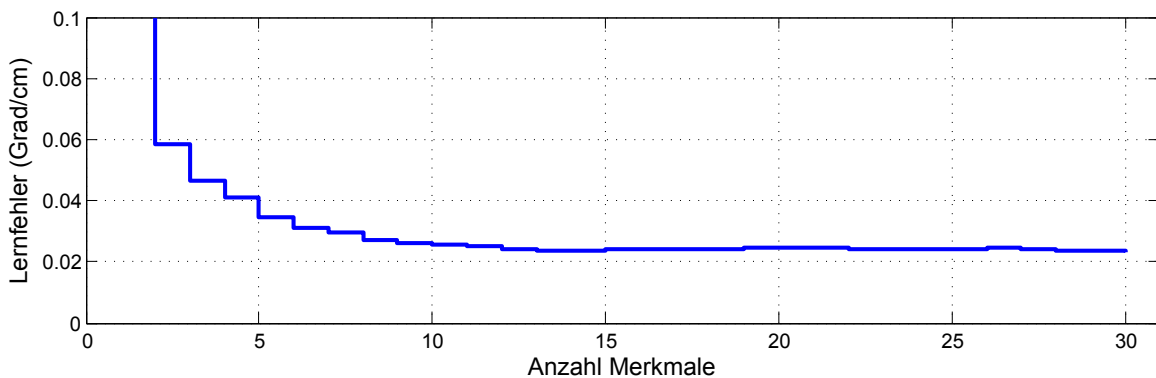


Abbildung 4.4: Lernfehler über den Verlauf der automatischen Merkmalsauswahl mittels FSS.

Zu sehen ist, dass der Lernfehler zu Beginn mit dem Hinzufügen weiterer Merkmale rapide abnimmt, bei größerer Anzahl Merkmale mit dem Hinzufügen weiterer Merkmale jedoch nur noch schwächer abnimmt oder sogar zunimmt. Das erste lokale Minimum des Lernfehlers liegt bei 13 Merkmalen und einem Lernfehler von  $0,024 \frac{\circ}{\text{cm}}$ . Es ist gleichzeitig das globale Minimum des Lernfehlers bei den durch FSS untersuchten Merkmalsvektoren.

### 4.3 Aufzeichnung von Trainingsdaten

Als Grundlage für instanzbasiertes Lernen müssen Trainingsinstanzen für die spätere Auswahl und Regression aufgezeichnet werden. Die Aufzeichnung produziert eine Sammlung von Instanzen, also Paaren von jeweils einem Merkmalsvektor  $\mathbf{x} = (x_1, x_2, \dots, x_n)$  und der reellen Zielgröße  $y$ .

Der Merkmalsvektor wird gemäß der beschriebenen Verfahren aus Kapitel 4.2.1 und 4.2.2 bestimmt. Die Ausgangsgröße ist gemäß Kapitel 4.1 die Bahnkrümmung.

Um Trainingsdaten zum Lernen eines reaktiven Roboterverhaltens aufzuzeichnen, wird der Roboter in der Form verfahren, die das zu lernende Verhalten zeigen soll. Während des Trainings werden in jedem Zeitschritt jeweils Sensordaten als Grundlage für den Merkmalsvektor



und Motorkommandos als Grundlage für die Zielgröße aufgezeichnet.

Im Rahmen dieser Arbeit werden dabei während der Aufzeichnung nicht Merkmalsvektor und Ausgangsgröße, sondern sämtliche Sensordaten und Motorkommandos aufgezeichnet. Dies vergrößert zwar den Speicherbedarf einer Aufzeichnung erheblich, erlaubt aber zum einen eine höhere Abtastrate, da die aufwendige Verarbeitung der Sensordaten zur Zeit der Aufzeichnung entfällt. Zum anderen erlaubt es die Neuberechnung der Instanzdatenbasis bei verändertem Merkmalsvektor oder bei Veränderung oder Ergänzung der Verfahren der Wahrnehmung und Merkmalsextraktion. Dies ist für ein Experimentalsystem von Vorteil.

Steht mit der Aufzeichnung von Roboterfahrten damit eine hinreichend große Instanzdatenbank zur Verfügung, kann der Roboter für jede Ausprägung des Merkmalsvektors nächste Nachbarn in der Instanzdatenbank suchen und durch Regression die geeignete Zielgröße zur aktuellen Situation bestimmen. Die Fahrten des Roboters zur Aufzeichnung der Trainingsdaten können dabei prinzipiell auf zwei Arten erfolgen: automatisch oder manuell.

#### 4.3.1 Manuelle Fahrt

Die manuelle Fahrt des Roboters kann über entsprechende Bediengeräte durch einen menschlichen Bediener vorgenommen werden. In dieser Arbeit fand die Aufzeichnung über Steuerung des Roboters mittels Joystick an einem, über das ARIA-Interface an den Roboter angeschlossenes, Notebook statt. Der Bediener kann so den Roboter steuern und die Sensordaten und Motorkommandos werden dabei aufgezeichnet.

Bei der manuellen Fahrt hat der Bediener direkte Kontrolle über das gelernte Roboterverhalten. Er kann bewusst Situationen auswählen und den Roboter nach seinen Vorstellungen steuern. Die resultierenden Aufzeichnungen repräsentieren das vom Bediener gewünschte Verhalten, sind jedoch nur bedingt reproduzierbar. Auf diese Art und Weise aufgezeichnete Fahrten können im Zweifelsfall zu einer mehrdeutigen Datenbasis führen, in der Instanzen gleicher oder sehr ähnlicher Merkmalsvektoren unterschiedliche Ausgangsgrößen aufweisen. Die Regression kann in diesem Fall fehlerhafte Werte produzieren.

Beispiel: Ein menschlicher Bediener wird sich bei einem in direkter Fahrtrichtung befindlichen Hindernis gegebenenfalls in einem Teil der Fälle für Ausweichen nach links, in einem anderen Teil der Fälle für ein Ausweichen nach rechts entscheiden. In der Instanzdatenbank führt dies zu Instanzen ähnlicher Merkmalsvektoren, aber grob unterschiedlicher Ausgangsgrößen. Eine mögliche Regression wäre eine Mittelung der Ausgangsgrößen der nächsten Nachbarn und damit in diesem Fall eine Geradeausfahrt, die zur Kollision mit dem Hindernis führt.

#### 4.3.2 Autonome Fahrt

Das Problem lässt sich reduzieren, indem das Verhalten nicht mit einer manuellen Fahrt, sondern einem autonomen Roboterverhalten trainiert wird. Dazu verfährt der Roboter autonom basierend auf seiner Sensorik und zeichnet dabei die Sensordaten und Motorkommandos auf. Das autonome Verhalten ist reproduzierbar und führt damit zu einer konsistenteren Daten-

basis als die Aufzeichnung manueller Fahrten.

Für die vorliegende Arbeit wurden Fahrten aufgezeichnet, die auf einem autonomen, kombinierten Verhalten für Hindernisvermeidung und Korridorfolge aufgrund der im *Pioneer 3-DX* verbauten Sonarsensoren basieren. Dafür wurde ein einfaches Verhalten implementiert, das die vorderen Sonarsensoren zur Kollisionsvermeidung nutzt und die seitlichen Sensoren zum Erkennen des Korridors und Ausrichten im Korridor. Zur Aufzeichnung der Daten wurde das Verhalten auf dem Roboter ausgeführt und gleichzeitig Sensordaten und Motorkommandos aufgezeichnet. Anschließend konnte Offline aus den Sensordaten und Motorkommandos jeden einzelnen Aufzeichnungsschrittes Merkmalsvektor und Zielgröße extrahiert werden.

Zu beachten ist in diesem Fall, dass auch die Aufzeichnung der Trainingsdaten mit dem autonomen Verhalten basierend auf Sonarsensorik zu einer mehrdeutigen Instanzdatenbasis führen kann. Das Verhalten ist zwar hinsichtlich der von den Sonarsensoren aufgenommenen Daten eindeutig, die Instanzen der Trainingsdaten werden allerdings auf Basis der bildgebenden Sensorik erzeugt. Gleiche Messungen der Sonarsensoren bedingen nicht zwangsläufig gleiche Merkmale bei der bildgebenden Wahrnehmung. Szenen, die die gleichen bildbasierten Merkmale erzeugen, müssen umgekehrt nicht eine gleiche Messung der Sonarsensoren bedeuten. Eine mehrdeutige Datenbasis muss bei der Modellbildung also trotzdem berücksichtigt werden.

## 4.4 Modellbildung

Die Modellbildung bei *Lazy Learning* findet zur Abfragezeit statt. Dazu steht eine Instanzdatenbank zur Verfügung, deren Instanzen aus Paaren von Merkmalsvektor  $\mathbf{x} = (x_1, x_2, \dots, x_n)$  und der reellen Zielgröße  $y$  bestehen. Der Merkmalsvektor wurde entsprechend Kapitel 4.2.2 aus denjenigen Merkmalen zusammengesetzt, die den Lernfehler minimierten. Die 13 damit gewählten Merkmale spannen einen 13-dimensionalen Instanzraum auf, die Ausgangsgröße ist gemäß Kapitel 4.1 die Bahnkrümmung.

Ziel der Modellbildung ist es, in lokaler Umgebung des Abfragepunktes mittels Regression ein Modell zu erstellen, das die Umgebung korrekt nachbildet und damit der Abfrageinstanz einen korrekten Ausgangswert zuweist.

Die Modellbildung muss, wie erwähnt, das Problem der Empfindlichkeit von instanzbasiertem Lernen gegenüber Rauschen berücksichtigen. Außerdem besteht bei maschinellem Lernen, insbesondere bei der Kollisionsvermeidung, das Problem der zu starken Glättung oder Generalisierung, was zu Fehlern führen kann. Abbildung 4.5 zeigt ein solches Beispiel. Der Roboter lernt in mehreren Instanzen, ein Hindernis rechts oder links zu umfahren (Trajektorien  $t_0, t_1, \dots, t_4$ ). Eine Glättung oder Generalisierung, die die aufgezeichneten Lösungen des Problems mittelt, führt zu einer Geradeausfahrt (Trajektorie  $t^*$ ) und damit zu einer Kollision des Roboters mit dem Hindernis. Dieser Fall muss vermieden werden.

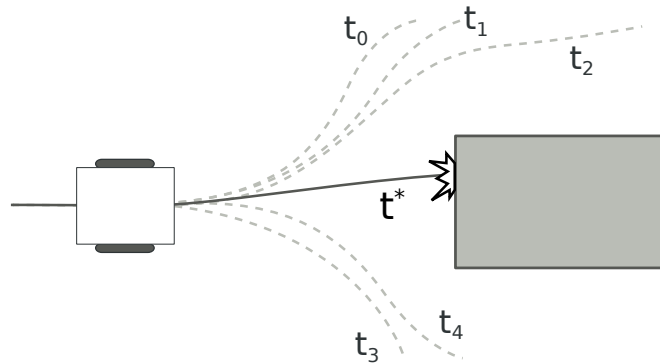


Abbildung 4.5: Roboterfahrt in der Draufsicht: Trainierte Trajektorien  $t_0, t_1, \dots, t_4$  und fehlerhaft gemittelte Trajektorie  $t^*$ .

#### 4.4.1 Regression

Mittels lokaler Regression wird in der Instanzdatenbank die Ausgangsgröße zu einer Abfrageinstanz ermittelt. Dazu werden zunächst über die Distanzfunktion die  $k$  nächsten Nachbarn, also die Instanzen, die der Abfrageinstanz am ähnlichsten sind, ermittelt.

In den gefundenen nächsten Nachbarn wird durch lokale Regression das Modell angenähert. Die hier genutzte *Lazy Learning Toolbox* erlaubt die Wahl eines konstanten, eines linearen, eines quadratischen Modells oder die automatische Kombination und Auswahl des besten Modells für die Modellbildung.

Mit den aufgezeichneten Daten mehrerer Roboterfahrten werden die genannten Modelle und die automatische Kombination getestet und validiert. Ergebnisse dazu finden sich in Kapitel 5.2.

#### 4.4.2 Clustering

Die Abfrage der Instanzdatenbank mit der aktuellen Abfrageinstanz gibt die  $k$  nächsten Nachbarn zurück. Anhand der  $k$  nächsten Nachbarn wird mittels lokaler Regression ein Modell erzeugt (siehe Kapitel 2.3.2). Lokale Regression mit den Modellen der Toolbox ist jedoch zum einen empfindlich gegenüber Rauschen, da Ausreißer die Regression mitunter stark beeinflussen. Zum anderen kann das Problem der in Abbildung 4.5 dargestellten fehlerhaften Glättung auftreten.

Zur Lösung beider Probleme erfolgt im Folgenden unter den gesuchten  $k$  nächsten Nachbarn nicht unmittelbar Regression, sondern Ausreißer und konkurrierende, das heißt stark abweichende, Ausgangsgrößen unter den nächsten Nachbarn, werden eliminiert. Um dies zu erreichen, werden mittels Clustering unter den  $k$  nächsten Nachbarn Gruppen von Instanzen gesucht, deren innere Distanz der Ausgangsgrößen  $\kappa$  einen vorgegebenen Grenzwert  $\theta_\kappa$  nicht überschreitet. Ergebnis des Clustering sind mehrere Cluster mit Instanzen jeweils ähnlicher Bahnkrümmung (*ähnlich* in Abhängigkeit von dem vorgegebenen Grenzwert  $\theta_\kappa$ ).

Wird nun die Regression zur Ermittlung der Bahnkrümmung nicht auf allen  $k$  nächsten Nachbarn, sondern nur auf dem größten der erhaltenen Instanz-Cluster ausgeführt, findet die Mittelung nur auf Instanzen ähnlicher Ausgangsgröße statt. Die Abweichung der Ausgangsgrößen der gespeicherten Instanzen und damit der maximale Fehler durch die Regression beträgt damit maximal die Hälfte des vorgegebenen Grenzwertes  $\theta_\kappa$ .

Dadurch, dass nur der größte Instanz-Cluster, also der Cluster mit den meisten hinsichtlich der Bahnkrümmung übereinstimmenden Instanzen, zur Regression verwendet wird, werden zudem Ausreißer und Rauschen eliminiert. Der vorangehenden Forderung, dies als Problem instanzbasierten Lernens zu berücksichtigen, ist damit entsprochen.

Zu beachten ist hierbei, dass das Clustern zu jedem Zeitschritt erneut geschieht. In der in Abbildung 4.5 dargestellten Situation ist demnach denkbar, dass bei Zufahrt auf das Hindernis der größte Cluster nicht immer die gleiche Richtung vorgibt, sondern zwischen den Zeitschritten von Instanzen, die ein Vorbeifahrt nach links vorgeben und Instanzen, die eine Vorbeifahrt nach rechts vorgeben, wechselt. In der Regel bedeutet allerdings eine einmalige Richtungsentscheidung, dass im nächsten Zeitschritt die Instanzen der gleichen Richtungsangabe erneut und deutlicher überwiegen, da sich dann das Hindernis bereits nicht mehr in direkter Fahrtrichtung, sondern bereits etwas seitlich der Fahrtrichtung des Roboters befindet.

## 4.5 Validierung

Zur Bewertung des maschinellen Lernens müssen die Lernergebnisse validiert werden. Die Validierung des Verhaltens erfolgt bei instanzbasierten Lernverfahren vornehmlich mittels eines Trainings- und eines Testdatensatzes von Instanzen. Das System wird mit der Datenbasis des Trainingsdatensatzes trainiert, anschließend wird der Lernfehler auf einem Testdatensatz mit Instanzen der gleichen Problemklasse ermittelt. Für die Instanzen des Testdatensatzes sind die richtigen Ausgangsgrößen bekannt.

Der Lernfehler wird bestimmt über die Distanz zwischen den trainierten Ausgangswerten der einzelnen Testinstanzen und den Ergebnissen des Lernprozesses. Der Gesamt-Lernfehler des Systems ist dann ein Maß über die Lernfehler der einzelnen Instanzen.

Die Trennung von Trainings- und Testset zur Validierung von Lernverfahren stellt sicher, dass tatsächlich ein allgemeines Modell gelernt wird und nicht die einzelnen Beispiele des Trainingssets nachgebildet werden.

Bei der Validierung instanzbasierten Lernens kann zum Beispiel eine Instanzdatenbank in einen Trainingsdatensatz und einen Testdatensatz aufgetrennt werden. Die Instanzen der beiden Datensätze sind damit aus der selben Problemklasse, sind aber nicht identisch. In dieser Arbeit wurde für die Validierung aus den Trainingsdaten der Roboterfahrten ein Drittel der Instanzen als Testdatensatz gezogen, zwei Drittel verblieben als Trainingsdatensatz. Bei der Aufteilung wurde darauf geachtet, dass die Instanzen des Trainings- und Testdatensatzes gleichverteilt über die Dauer der Roboterfahrt sind, um alle vorkommenden Situationen der

Fahrt in die Validierung mit einzubeziehen.

Um außerdem die Forderung nach der Robustheit des Verfahrens zu überprüfen, wurden anschließend Trainings- und Testdatensätze aus unterschiedlichen Roboterfahrten in unterschiedlichen Umgebungen in der Art kombiniert, dass die Instanzen des Testsets aus einer anderen Umgebung stammen als die Instanzen des Trainingssets. Ergebnisse dazu finden sich in Kapitel 5.3.



# Kapitel 5

## Experimentelle Ergebnisse

Um die Robustheit des Wahrnehmungsapparates und des darauf aufbauenden, trainierte Roboterhaltens zu testen, wurden Experimente mit dem realen Robotersystem durchgeführt, die im Folgenden dargestellt werden.

Der mobile Roboter, der zum Testen der Software verwendet wurde, ist das in Kapitel 2.1 beschriebene System bestehend aus einem *Pioneer 3-DX*, erweitert um zusätzliche, bildgebende Sensorik. Als Sensorik wird die 3D-Kamera *O3d* von *PMD Technologies* mit einer Auflösung von  $64 \times 50$  Pixeln und einer Reichweite von  $0 - 7,50\text{ m}$  verwendet, sowie eine monokulare USB-Kamera mit einer Auflösung von  $640 \times 480$  Pixeln und eine Kamera mit omnidirektionaler Optik. Bei dieser Kamera handelt es sich um eine Firewire-Kamera mit einer Auflösung von  $1280 \times 960$  Pixeln, ausgestattet mit einem katadioptrischen System, das eine Rundumsicht von  $360^\circ$  mit einem vertikalen Öffnungswinkel von  $15^\circ$  über dem Horizont und  $60^\circ$  unter dem Horizont ermöglicht. Die Optik bildet die Rundumsicht auf dem Kamerasensor auf einen kreisförmigen Bereich mit  $960$  Pixeln Durchmesser ab. Die Anordnung der Kameras ist wie in Abbildung 3.1 dargestellt, mit einem Nickwinkel der monokularen Kamera und der 3D-Kamera gegen den Horizont von  $35^\circ$ .

Aufbauend auf dieser Hardware-Konfiguration des Roboters wurde der in den obigen Kapiteln dargestellte Wahrnehmungsprozess und das Erzeugen des Roboterhaltens in der Entwicklungssoftware *Matlab* implementiert. Im Folgenden wird die Robustheit der Wahrnehmung untersucht und im Anschluss die Fähigkeit des maschinellen Lernprozesses, das trainierte Verhalten mit bildbasierter Sensorik nachzubilden.

### 5.1 Robustheit der Wahrnehmung

Der Aufbau des Wahrnehmungsprozesses ist in Abbildung 5.1 dargestellt. Der linke Teil der Abbildung stellt die Verarbeitung der 3D-Daten der PMD-Kamera dar. Die Segmentierung und Klassifizierung wird anschließend zur Verbesserung der Segmentierung in der monokularen Ansicht sowie als Grundlage zur Bewertung der Bildsegmentierungsverfahren verwendet.

Die Segmentierung der monokularen Ansicht ist Grundlage für die Extraktion einiger Merkmale, die Bewertung der Bildsegmentierungsverfahren dient der Auswahl der Bildsegmentie-

rungsverfahren für die omnidirektionale Ansicht. Die dort erfolgte Segmentierung dient der Extraktion weiterer Merkmale zur Repräsentation der lokalen Umgebung des Roboters.

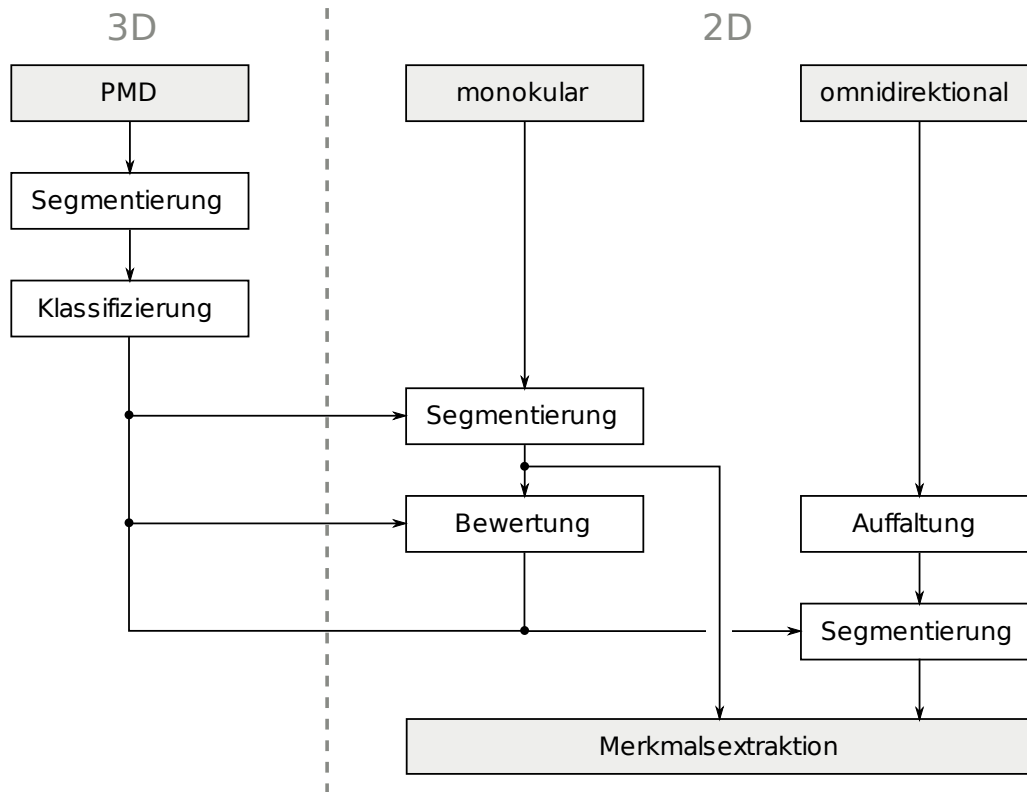


Abbildung 5.1: Aufbau des Wahrnehmungsprozesses.

Die Robustheit der Wahrnehmung wird in verschiedenen Beispielszenen getestet und mit Ergebnissen herkömmlicher bildbasierter Wahrnehmung verglichen. Außerdem werden ausgesuchte Situationen untersucht, in denen herkömmliche Bodensegmentierung zu *falsch Positiven* oder *falsch Negativen* führt. In dieser Anwendung bedeutet dies:

1. **Falsch Positive:** Die Erscheinung eines Hindernisses unterscheidet sich nicht deutlich von der des Bodens (zum Beispiel gleiche Farbe, gleiche oder keine Textur). Herkömmliche Bildsegmentierung ordnet das Hindernis mangels Unterscheidungsmerkmalen fälschlicherweise dem Boden zu und klassifiziert sie daher als befahrbare Fläche. Das hier entwickelte System soll in der Lage sein, das Hindernis unter der Nutzung der 3D-Kamera trotzdem korrekt zu erkennen und zu klassifizieren.
2. **Falsch Negative:** Die Erscheinung eines Bodensegments weicht deutlich von der Erscheinung des übrigen Bodens ab (z. B. in Farbe oder Textur). Dies kann zum Beispiel ein auf dem Boden liegendes Papier sein, das sich in der Farbe von der des Bodens unterscheidet. Herkömmliche Bildsegmentierung könnte das Blatt nicht dem Boden zurechnen und würde es als Hindernis merkieren. Das hier entwickelte System soll in der



Lage sein, das Bodensegment korrekt als befahrbare Bodenfläche zu klassifizieren.

### 5.1.1 Falsch Positive

*Ein Hindernis wird fälschlicherweise als Boden, und damit als für den Roboter befahrbar klassifiziert.*

Die folgenden beiden Ansichten zeigen Beispielszenen, in denen falsch Positive auftreten können.

Abbildung 5.2 zeigt eine konstruierte Szene, in der ein Hindernis mit der Textur des Bodens beklebt wurde und damit in Farbe und Textur dem Boden gleicht. Das Hindernis ist in der Abbildung zur besseren Sichtbarkeit durch eine schwarze Kontur hervorgehoben. Die mittlere Ansicht zeigt die Segmentierung des Bodens mittels *Region Growing* und einem einzelnen Startpunkt in der unteren Bildmitte. Aufgrund mangelnder Unterscheidbarkeit von Boden und Hindernis, wird das Hindernis zu großen Teilen dem Bodensegment zugerechnet. Die rechte Ansicht zeigt das Ergebnis der Segmentierung mittels *Marker-based Watershed* mit Markern aus der 3D-Klassifizierung. Hierbei wird eine Klassifizierung des Hindernisses als Boden dadurch verhindert, dass der Bildbereich des Hindernisses über die 3D-Klassifizierung bereits als Hindernis bekannt ist und dem Segmentierungsverfahren als dem entsprechender Marker übergeben wird.

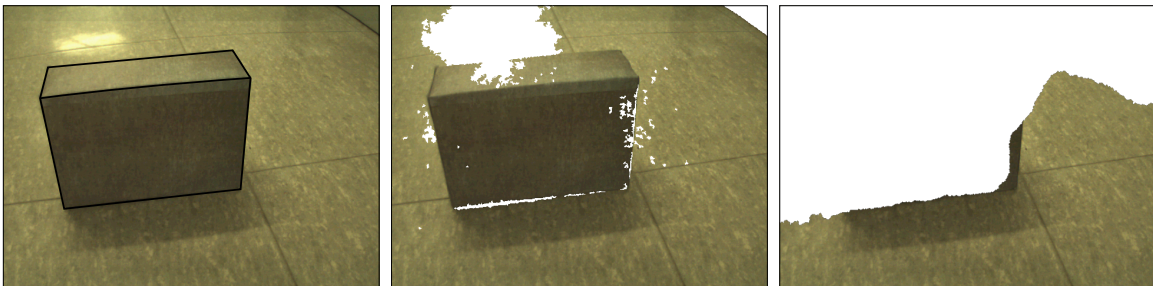


Abbildung 5.2: Vergleich der Bodensegmentierung in einer Beispielszene. Originalansicht (links), Segmentierung mit *Region-Growing* und einzelнем Startpunkt (Mitte) und *Marker-based Watershed* mit Markern aus der 3D-Klassifizierung.

Abbildung 5.3 zeigt eine Szene, in der der Roboter direkt vor einer Wand steht. Herkömmliche Segmentierung, mit einem Startpunkt in der unteren Bildmitte schlägt hier aufgrund einer falschen Anfangshypothese fehl. Dadurch werden segmentiert Teile der Ansicht (eigentlich Wand) als Boden und damit als befahrbare Fläche gekennzeichnet. Bei der Segmentierung mit Startpunkten aus der 3D-Klassifizierung ist bekannt, dass sich kein Boden im Sichtbereich befindet, dementsprechend erfolgt mangels Startpunkten keine Bodensegmentierung, der gesamte Bildbereich wird stattdessen als „Wand“ markiert.

Der Vergleich der herkömmlichen Segmentierung mit der robusten Segmentierung basierend auf der 3D-Messung gemäß den Bewertungskriterien aus Kapitel 3.3, in diesem Fall beschränkt



Abbildung 5.3: Vergleich der Bodensegmentierung in einer Beispielszene. Originalansicht (links), Segmentierung mit *Region-Growing* und individuellem Startpunkt (Mitte) und *Region-Growing* mit Startpunkten aus der 3D-Klassifizierung.

auf die Auswertung der falsch Positiven, belegt die Verringerung der Anzahl an Fehlzuordnungen durch das entwickelte Verfahren, wie Tabelle 5.1 zeigt. *Region-Growing* erfolgt hier unter anderem mit Startpunkten aus der 3D-Klassifizierung, der Einfachheit halber als „klassifizierte Startpunkte“ bezeichnet.

	Segmentierungsverfahren	Anzahl falsch Positiver	$b_{fp}$
Abbildung 5.2	<i>Region-Growing</i> (1 Startpunkt)	1235 von 1348	0,92
	<i>Marker-Based Watershed</i>	40 von 674	0.06
Abbildung 5.3	<i>Region-Growing</i> (1 Startpunkt)	2564 von 3121	0,82
	<i>Region-Growing</i> (klassifizierte Startpunkte)	0 von 1720	0,00

Tabelle 5.1: Vergleich der Bildsegmentierungsverfahren bezüglich falsch Positiver.

### 5.1.2 Falsch Negative

*Befahrbare Bodenfläche wird fälschlicherweise nicht als Boden, sondern als nicht-befahrbar klassifiziert.*

Die Ansichten in Abbildung 5.4 zeigen eine Szene, in denen die Bodenfläche heterogen ist und sich in einzelnen Bildbereichen in ihrer Erscheinung unterscheidet. In diesem Fall durch ein auf dem Boden liegendes Bogen Papier, der sich in Farbe und Textur deutlich von dem Untergrund unterscheidet. Links ist die unsegmentierte monokulare Ansicht dargestellt. Die Ansicht in die Mitte der Abbildung zeigt das Ergebnis einer Segmentierung mit *Region-Growing*, basierend auf einem Startpunkt in der Mitte des unteren Bildrands. Die Bilder rechts zeigen das Ergebnis der verbesserten Segmentierung mit *Region-Growing* basierend auf Startpunkten aus der 3D-Klassifizierung.

Zu sehen ist, dass die Bodensegmentierung basierend auf Startpunkten aus der 3D-Klassifizierung deutlich mehr Bodenfläche korrekt segmentiert als die herkömmliche Segmentierung mit nur einem Startpunkt. Durch die weiter gestreuten Startpunkte liegen Startpunkte beiderseits großer Gradienten im Bild, wodurch große Gradienten in der Intensität des Bodens übersprungen werden. Dadurch werden selbst Bildbereiche mit starker Reflexion korrekt als

Bodenfläche segmentiert.



Abbildung 5.4: Vergleich der Bodensegmentierung in einer Beispielszene. Originalansicht (links), Segmentierung mit *Region-Growing* und einzelнем Startpunkt (Mitte) und *Region-Growing* mit Startpunkten aus der 3D-Klassifizierung.

Der Vergleich der herkömmlichen Segmentierung mit der robusten Segmentierung, hier beschränkt auf die Auswertung der falsch Negativen, belegt die Verringerung der Anzahl an Fehlzuordnungen durch das entwickelte Verfahren, wie Tabelle 5.2 zeigt. Die unterschiedliche Anzahl der gesamt-möglichen falsch Negativen (zweite Spalte der Tabelle) erklärt sich dadurch, dass für die beiden Verfahren unterschiedlich viele Startpunkte benutzt werden und dementsprechend unterschiedlich viele – nämlich die restlichen – klassifizierten 3D-Punkte zur Überprüfung der falsch Negativen übrig bleiben.  $b_{fn}$  (dritte Spalte der Tabelle) zeigt daher die relative Anzahl an falsch Negativen.

	Segmentierungsverfahren	Anzahl falsch Negative	$b_{fn}$
Abbildung 5.4 a)	<i>Region-Growing</i> (1 Startpunkt)	1088 von 3073	0.35
	<i>Region-Growing</i> (klassifizierte Startpunkte)	3 von 2048	0.00

Tabelle 5.2: Vergleich der Bildsegmentierungsverfahren bezüglich falsch Negativer.

### 5.1.3 Fazit

Die Experimente zeigen, dass das in Kapitel 3 vorgestellte Vorgehen geeignet ist, die Bildsegmentierung zur Findung der Bodenfläche zu verbessern und die Robustheit gerade in Grenzsituationen zu erhöhen.

Die konstruierten Szenen zum Testen von falsch Positiven und falsch Negativen zeigen, dass der Forderung nach Robustheit entsprochen ist. Falsch Negative werden dadurch verringert, dass durch die 3D-Klassifizierung die Anfangshypothese für die Bildsegmentierung besser und ausführlicher ist. Bei *Marker-based Watershed* werden die falsch positiven Zuordnungen dadurch verringert, dass die Segmentierung einzelner Bildbereich dadurch verhindert wird, dass Bildbereiche durch die 3D-Klassifizierung bereits explizit als Wand oder Hindernis bekannt und gekennzeichnet sind.

Insbesondere die Vermeidung von falsch Positiven ist dabei für die Roboternavigation in besonderem Maße bedeutend, da sie Beschädigungen des Roboters und der Umwelt sowie, in Umgebung von Menschen, Verletzungen verhindern kann.

## 5.2 Roboterverhalten

Das Roboterverhalten wird anhand seiner Fähigkeit bewertet, das trainierte Verhalten nachzubilden. Dazu werden einzelne Situationen sowie der allgemeine Lernfehler betrachtet. Gemäß den Ausführungen aus Kapitel 4 wird das Roboterverhalten dazu auf der Basis von Trainingsdaten durch instanzbasiertes Lernen trainiert. Abbildung 5.5 zeigt den Ablauf des maschinellen Lernens des Roboterhaltens.

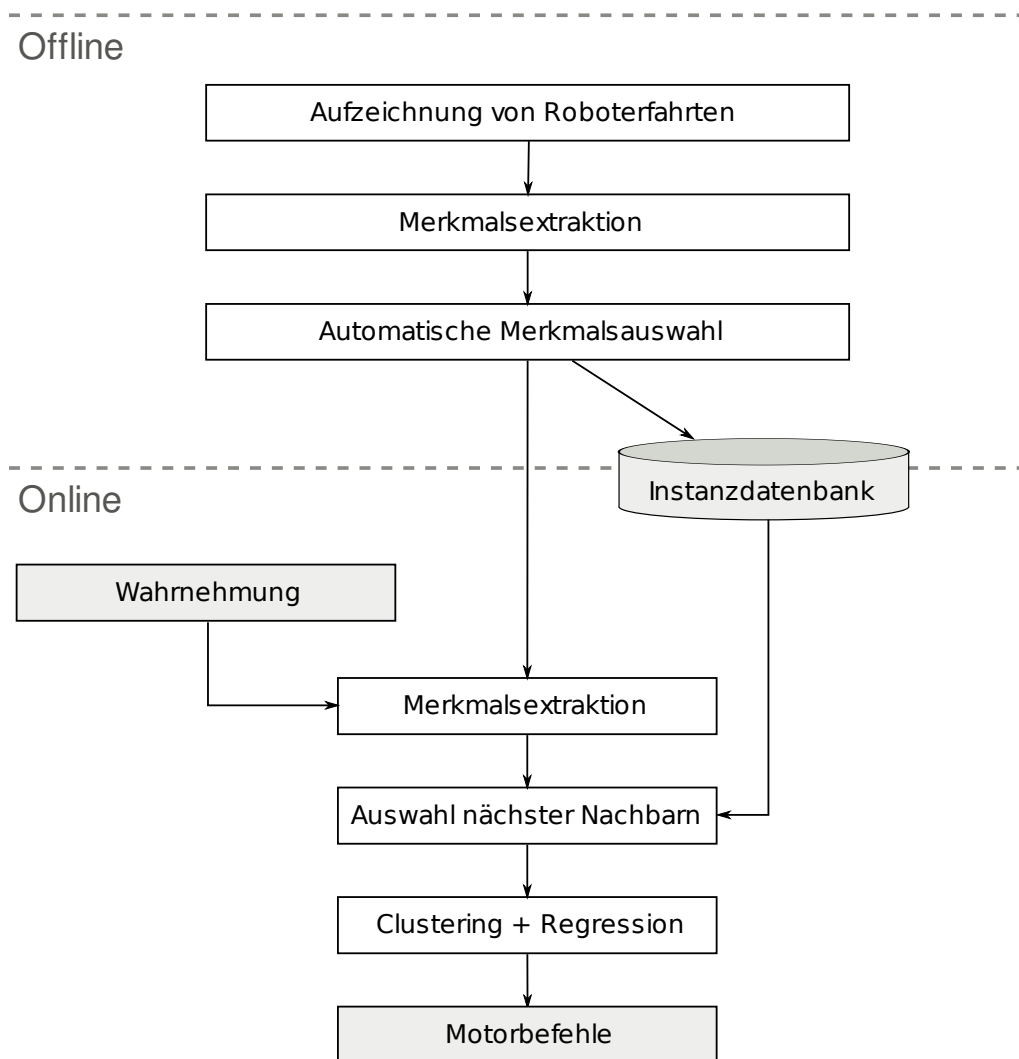


Abbildung 5.5: Schematischer Ablauf des maschinellen Lernens des Roboterhaltens.

Zu Beginn werden Trainingsdaten als Grundlage für die Instanzdatenbank aufgezeichnet. Die autonomen Roboterfahrten basieren auf einem einfachen, programmierten Roboterverhalten zur gleichzeitigen Kollisionsvermeidung und Korridorfolge, basierend auf den Sonarsensoren des *Pioneer 3-DX*. Für die Aufzeichnung zeichnet der Roboter dreimal pro Sekunde ein komplettes Abbild aller drei Kameras, sowie die durch das Verhalten vorgegebene Rotationsgeschwindigkeit und Vorwärtsgeschwindigkeit auf.

Auf der Aufzeichnung findet offline die Merkmalsextraktion statt, die gemeinsam mit der aus den Geschwindigkeiten ermittelten Bahnkrümmung Grundlage für die Instanzen bildet. Auf sämtlichen Daten werden mittels automatischer Merkmalsselektion die relevantesten Merkmale bestimmt und die Instanzen reduziert auf diese Merkmale in einer Instanzdatenbank abgelegt.

Zur Laufzeit des Roboterhaltens findet die Wahrnehmung und Merkmalsextraktion auf den Live-Aufnahmen der drei Kameras statt. Anhand der Ausprägung der Merkmale werden in der Instanzdatenbank die nächsten Nachbarn ausgewählt, in denen nach Clustern ähnlicher Bahnkrümmung gesucht wird und anschließend die lokale Modellbildung und Regression erfolgt.

Bei der Kollisionsvermeidung der autonomen Roboterfahrt basierend auf Sonarsensorik wurde darauf geachtet, dass der Roboter erst vor Hindernissen ausweicht, wenn diese sich auch im Sichtfeld der Front-Kameras des Roboters befinden, um ein Lernen der Abhängigkeit zwischen segmentierter Frontansicht und Reaktion des Roboters zu ermöglichen.

Für die folgenden Experimente wurden 1630 Szenen (Kamerabilder und Motorbefehle) aufgezeichnet, aus denen 1630 Instanzen (Merkmalsvektor und Ausgangsgröße) erzeugt wurden. Verglichen wird im Folgenden, wie sich einzelne Komponenten des Lernprozesses auf das resultierende Roboterverhalten, insbesondere den Lernfehler, auswirken. Die Ergebnisse werden in Abhängigkeit von Veränderungen des Lernverfahrens, wie Wahl der Merkmale, Clustern der nächsten Nachbarn und Wahl unterschiedlicher Lernmodelle, gesetzt.

Zuerst werden Ergebnisse eines Verhaltens dargestellt, das entsprechend Kapitel 4 erlernt wurde. Das Verhalten basiert auf automatischer Merkmalsauswahl durch *Forward Sequential Selection* auf Basis der 48 in Kapitel A aufgezeigten Merkmale. Zur lokalen Modellbildung wird in der *Lazy Learning Toolbox* das automatische Kombinieren und Auswählen des besten Modells eingestellt.

Nachdem aus den aufgezeichneten Szenen der Roboterfahrt die Merkmalsvektoren mit den gesamten 48 Merkmalen und die jeweils zugehörige Bahnkrümmung als Ausgangsgröße errechnet wurden, wurde die automatische Merkmalsselektion gestartet. Abbildung 5.6 zeigt die Entwicklung des Lernfehlers in der Bahnkrümmung in Abhängigkeit von der Anzahl der durch FSS gewählten Merkmale. Dazu wird zu jedem Merkmalsvektor das Lernverfahren gemäß Kapitel 4.5 validiert und der Lernfehler bestimmt. Der Lernfehler ist der durchschnittliche Fehler über alle Instanzen der Testdatenbank, also die durchschnittliche Abweichung zwischen trainierter und gelernter Bahnkrümmung, gegeben in  $\frac{\circ}{cm}$ .

Das erste lokale Minimum des Lernfehlers liegt bei 13 Merkmalen und einem Lernfehler von

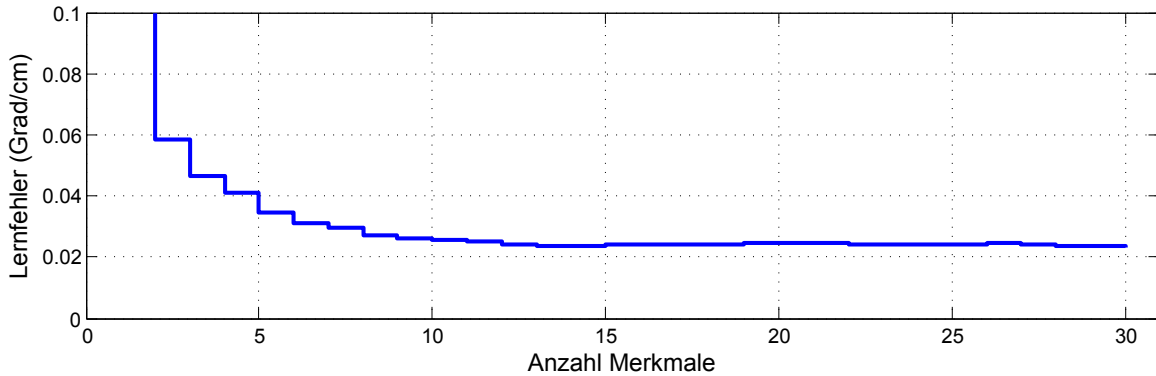


Abbildung 5.6: Lernfehler über den Verlauf der automatischen Merkmalsauswahl der ersten 30 Merkmale mittels FSS.

$0,24 \frac{\circ}{cm}$ . Dies entspricht einer Abweichung von ca. 2,5 Grad auf einem Meter Fahrt des Roboters. Nach dem ersten lokalen Minimums bleibt der Lernfehler weitestgehend konstant, weitere Merkmale verringern den Lernfehler nicht mehr signifikant oder vergrößern ihn sogar. Dass das Hinzufügen von Merkmalen den Lernfehler vergrößern kann, liegt in der Natur des instanzbasierten Lernens, da Merkmale, die keine Aussage über die Ähnlichkeit zweier Instanzen zulassen (also für die Zuordnung irrelevant sind) trotzdem zur Distanzfunktion beitragen und dadurch die Auswahl der nächsten Nachbarn beeinflussen.

### 5.2.1 Merkmalsauswahl

Im Folgenden wird betrachtet, wie sich die automatische Auswahl der Merkmale auf den Lernfehler und damit die Güte des Lernverfahrens auswirkt. Die automatische Auswahl ist zum einen notwendig, um irrelevante Merkmale zu vermeiden, die bei instanzbasierten Lernverfahren zu falschen Ergebnissen führen können, zum anderen ist zu erwarten, dass durch die Beschränkung auf die relevantesten Merkmale auch eine Verringerung des Lernfehlers eintritt.

Zur Überprüfung wird dafür der Lernfehler bei manueller Auswahl der Merkmale gemäß Kapitel 4.2.1 dem Lernfehler der automatischen Merkmalsauswahl gegenübergestellt. Mit den manuell gewählten 9 Merkmalen basierend auf Expertenwissen wird ein durchschnittlicher Lernfehler von  $0,039 \frac{\circ}{cm}$  erreicht. Abbildung 5.7 stellt die Entwicklung des Lernfehlers bei automatischer Merkmalsauswahl mittels FSS der manuellen Auswahl der Merkmale gegenüber. Gestrichelt ist der konstante Lernfehler der manuell gewählten 9 Merkmale aufgetragen, die durchgezogene Linie zeigt die Entwicklung des Lernfehlers bei der automatischen Merkmalsauswahl.

Zu sehen ist, dass ein automatisch gewählter Merkmalsvektor bereits mit 5 Merkmalen den Lernfehler des manuell gewählten 9-dimensionalen Merkmalsvektors unterschreitet. Das Minimum des Lernfehlers bei der automatischen Merkmalsauswahl liegt mit  $0,024 \frac{\circ}{cm}$  bei ca. 61% des Lernfehlers der manuellen Auswahl. Obwohl die Merkmale manuell mit Expertenwissen basierend auf vorigen Roboterverhalten gewählt wurden, sorgt die automatische Auswahl des

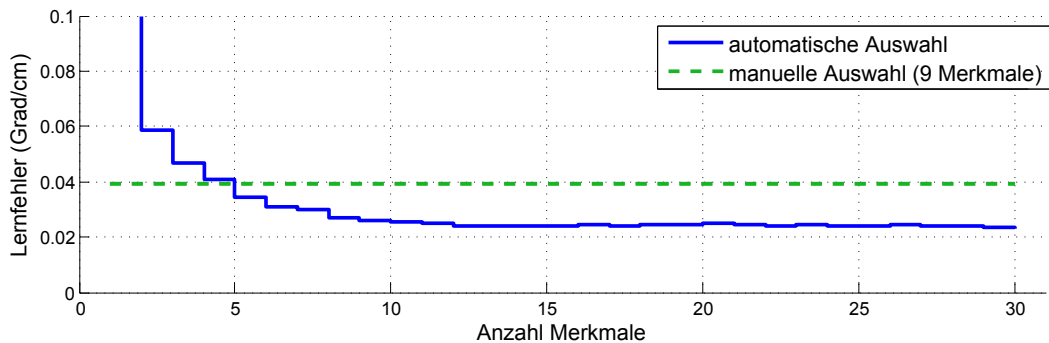


Abbildung 5.7: Vergleich des Lernfehlers der manuellen Merkmalsauswahl und der automatischen Merkmalsauswahl.

Merkmalsvektors also für eine signifikante Verbesserung des Lernergebnisses. Eine vergleichbare Güte des Lernvorgangs wird bereits mit 5 statt 9 Merkmalen erreicht und die selbe Anzahl automatisch gewählter Merkmale verringert den Lernfehler gegenüber den manuell gewählten Merkmalen auf  $0,026 \frac{\circ}{cm}$  und damit um ca. ein Drittel.

### Merkmalsvektor

Die Auswahl der relevantesten Merkmale wird mit seine Entwicklung bei unterschiedlichen Trainingsdaten untersucht. Betrachtet werden die durch FSS ausgewählten Merkmale über die gesamte Menge der aufgezeichneten Trainingsdaten bis zur Erreichung des globalen Minimums des Lernfehlers (13 Merkmale). 9 der automatisch ausgewählten Merkmale (ca. 70 %) sind aus der Frontansicht extrahierte Merkmale, 4 der 15 Merkmale sind aus der omnidirektionalen Ansicht extrahierte Merkmale. Das relevanteste Merkmal ist eines der omnidirektionalen Ansicht, gefolgt von einem Merkmal der Frontansicht.

Wird die automatische Auswahl der Merkmale auf Trainingsdaten ausgeführt, die in Situationen aufgezeichnet wurden, die fast ausschließlich Kollisionsvermeidung zeigen, sind die automatisch ausgewählten Merkmale ähnlich zusammengesetzt. Bis zum Erreichen des ersten lokalen Minimums des Lernfehlers bei 12 Merkmalen, wurden 9 Merkmale (75 %) aus der Frontansicht gewählt, die verbleibenden 3 Merkmale aus der omnidirektionalen Ansicht. Die ersten beiden, also die relevantesten beiden Merkmale, sind aus der Frontansicht extrahierte Merkmale.

Wird die automatische Auswahl der Merkmale auf Trainingsdaten ausgeführt, die in Situationen aufgezeichnet wurden, die fast ausschließlich Korridorfolge zeigen, sind zum Erreichen des ersten lokalen Minimums des Lernfehlers mit 13 Merkmalen wiederum 9 Merkmale (ca. 70 %) aus der Frontansicht. Die verbleibenden 4 Merkmale sind aus der omnidirektionalen Ansicht extrahierte Merkmale. Die relevantesten beiden Merkmale sind aus der omnidirektionalen Ansicht extrahiert.

Tabelle 5.2.1 zeigt die Ergebnisse in einer Übersicht. Aufgeführt ist jeweils die Anzahl  $f_n$  der

Merkmale bis zum Erreichen des ersten lokalen Minimum, die Anzahl  $f_{mono}$  der Merkmale aus der Frontansicht und die Anzahl  $f_{omni}$  der Merkmale aus der omnidirektionalen Ansicht bei verschiedenen Trainingsdaten.

<b>Trainingsdaten</b>	$f_n$	$f_{mono}$	$f_{omni}$
Kollisionsvermeidung	12	9	3
Korridorfolge	13	9	4
Gemischt	13	9	4

Tabelle 5.3: Übersicht über die Merkmalsauswahl bei verschiedenen Trainingsdaten.

Zu sehen ist, dass der Merkmalsvektor zum Erreichen des ersten lokalen Minimums bei allen Trainingsdaten in ähnlichem Verhältnis aus Merkmalen der omnidirektionalen Ansicht und der Frontansicht zusammengesetzt ist. Dabei ist jedoch die Relevanz der Merkmale unterschiedlich. Zeigen die Trainingsdaten überwiegend Kollisionsvermeidung, sind im Verhältnis zu gemischten Situationen als Trainingsdaten die Merkmale der Frontansicht relevanter, werden also bei der automatischen Merkmalsauswahl früher gewählt. Dies ist insofern plausibel, dass der Bodenbereich in Fahrtrichtung unmittelbar vor dem Roboter, der für die Kollisionsvermeidung relevant ist, in der Frontansicht sehr viel hochauflösender segmentiert und repräsentiert wird als in der omnidirektionalen Ansicht.

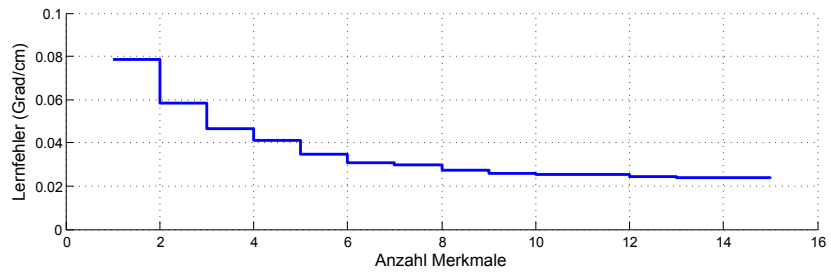
Auf Trainingsdaten, die überwiegend Korridorfolge zeigen, werden dahingegen gegenüber den gemischten Trainingsdaten diejenigen Merkmale früher gewählt, sind also relevanter, die die omnidirektionale Ansicht repräsentieren. Dies ist plausibel, da die Geometrie des Korridors in der monokularen Ansicht nicht oder nur bedingt sichtbar ist.

### 5.2.2 Lernmodell

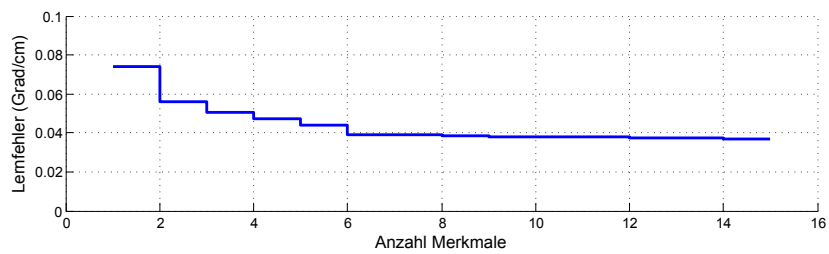
Die *Lazy Learning Toolbox* lässt die Wahl des Modells für die lokale Modellbildung, zu. Dabei kann zwischen einem konstanten Modell, einem linearen Modell, einem quadratischen Modell und der automatischen Auswahl der besten Kombination aus den Modellen gewählt werden (siehe Erläuterungen in Kapitel 2.3.2). In Abbildungen 5.8 werden die verschiedenen Modelle einander gegenüber gestellt. Aufgetragen ist hier jeweils nur die Auswahl der ersten 15 Merkmale mittels FSS. Dabei kann anhand des Lernfehlers beobachtet werden, welches Modell sich für die vorliegende Aufgabe eignet.

Überraschenderweise sind das lineare Modell und das quadratische Modell hinsichtlich des globalen Minimums des Lernfehlers schlechter als das konstante Modell, obwohl sie theoretisch eine genauere Approximation erlauben. Das globale Minimum bei Nutzung des linearen Modells liegt bei  $0,038 \frac{\circ}{cm}$ , bei Nutzung des quadratischen Modells bei  $0,039 \frac{\circ}{cm}$ . Bei Nutzung des konstanten Modells liegt das globale Minimum mit  $0,024 \frac{\circ}{cm}$  deutlich niedriger. Die Verringerung des Lernfehlers bei steigender Anzahl Merkmale verläuft darüber hinaus bei dem linearen und quadratischen Modell flacher als bei dem konstanten Modell, der Lernfehler verringert sich also nicht so schnell wie bei Nutzung des konstanten Modells.

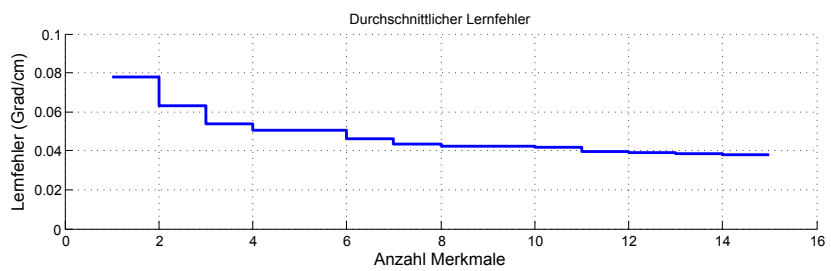




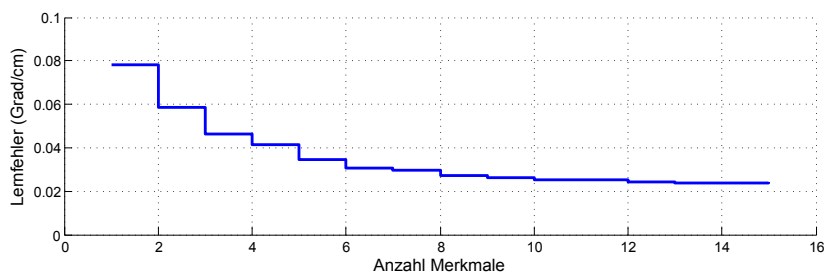
(a) Konstantes Modell



(b) Lineares Modell



(c) Quadratisches Modell



(d) Kombiniertes Modell

Abbildung 5.8: Entwicklung des Lernfehlers bei verschiedenen Lernmodellen.

Der Verlauf des Lernfehlers bei Nutzung der automatische Auswahl der besten Kombination der Modelle ist dem Verlauf des Lernfehlers bei Nutzung des konstanten Modells nahezu identisch. Bei der automatischen Auswahl des Modells zur Laufzeit wird fast ausschließlich das konstante Modell bevorzugt. Der Rechenaufwand des kombinierten Modells ist dabei jedoch gegenüber dem konstanten Modell stark erhöht, da – gerade mit steigender Anzahl der Merkmale – eine Menge von Kombination verschiedener Modelle ausgetestet wird. Die Wahl des Lernmodells fällt daher auf das einfachere und schnellere konstante Modell, eine automatische Auswahl des Modells zur Laufzeit bringt keinen signifikanten Gewinn.

### 5.2.3 Clustering

Zur Eliminierung von Ausreißern und Verhinderung einer fehlerhaften Glättung, im Zuge der lokalen Regression, wird, wie in Kapitel 4.4 dargestellt, unter den gewählten nächsten Nachbarn der Abfrageinstanz hinsichtlich der Ausgangsgröße geclustert. Die Regression wird dann lediglich auf den Nachbarn des größten Clusters ausgeführt.

Im Folgenden wird untersucht, inwiefern dies Auswirkung auf die Güte des Lernprozesses hat. Abbildung 5.9 zeigt die Entwicklung des Lernfehlers bei automatischer Merkmalsauswahl mit und ohne Clustering.

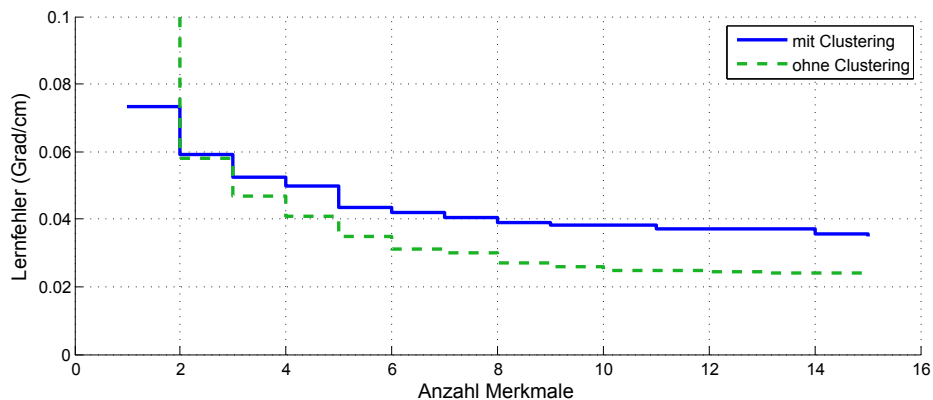


Abbildung 5.9: Vergleich der Entwicklung des Lernfehlers mit und Clustering.

Zu sehen ist, dass der Lernfehler durch das Clustern gegenüber der Regression ohne Clustering steigt. Dies lässt sich dadurch erklären, dass durch das Clustering einige Instanzen aus der Umgebung der Abfrageinstanz bei der Modellbildung ignoriert werden, zur Bestimmung des Lernfehlers aber weiterhin beitragen.

## 5.3 Gesamtsystem

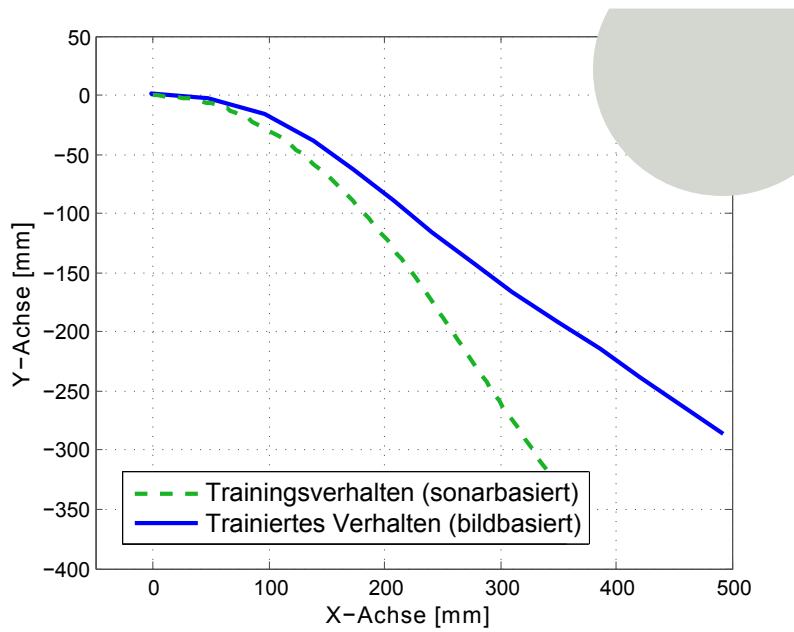
Das Gesamtsystem, bestehend aus Wahrnehmungsapparat und gelerntem Roboterverhalten, wird hinsichtlich seiner Robustheit in verschiedenen Innenraum-Umgebungen untersucht. Die Robustheit des Systems ist dabei weitestgehend durch die Robustheit der Wahrnehmung gegeben.

Um das Gesamtsystem zu bewerten, werden Trainingsdaten in verschiedenen Umgebungen aufgezeichnet und das damit trainierte Roboterverhalten betrachtet. Zuerst wird das komplette System mit der Wahrnehmung gemäß Kapitel 3 und dem gelernten Verhalten gemäß Kapitel 4 insgesamt auf seine Fähigkeit untersucht, ein trainiertes Verhalten nachzubilden. Ergänzend zu den Ergebnissen bezüglich des resultierenden Lernfehlers (dargestellt in den vorigen Kapiteln), wird die Trajektorie einer Fahrt des Roboters mit dem auf Sonarsensorik basierenden Trainingsverhalten aufgezeichnet und anschließend mit der gefahrenen Trajektorie des in dieser Arbeit entwickelten trainierten Roboterverhaltens verglichen. Dazu wurden jeweils die Trajektorien mit den Odometrie-Sensoren des Roboters aufgezeichnet und anschließend in Deckung gebracht. Abbildung 5.10 zeigt den Vergleich zweier Fahrten. Dabei sind deckungsgleiche Trajektorien der beiden Fahrten aufgrund unterschiedlicher Sensorik und einer Mittelung im Rahmen des maschinellen Lernens nicht zu erwarten. Bewertet werden kann hier aber die grundsätzliche Übereinstimmung des Verhaltens anhand ähnlicher Bewegungen in unterschiedlichen Situationen. Aufgrund der Ungenauigkeit der Odometriesensorik wurden nur kurze Strecken gefahren.

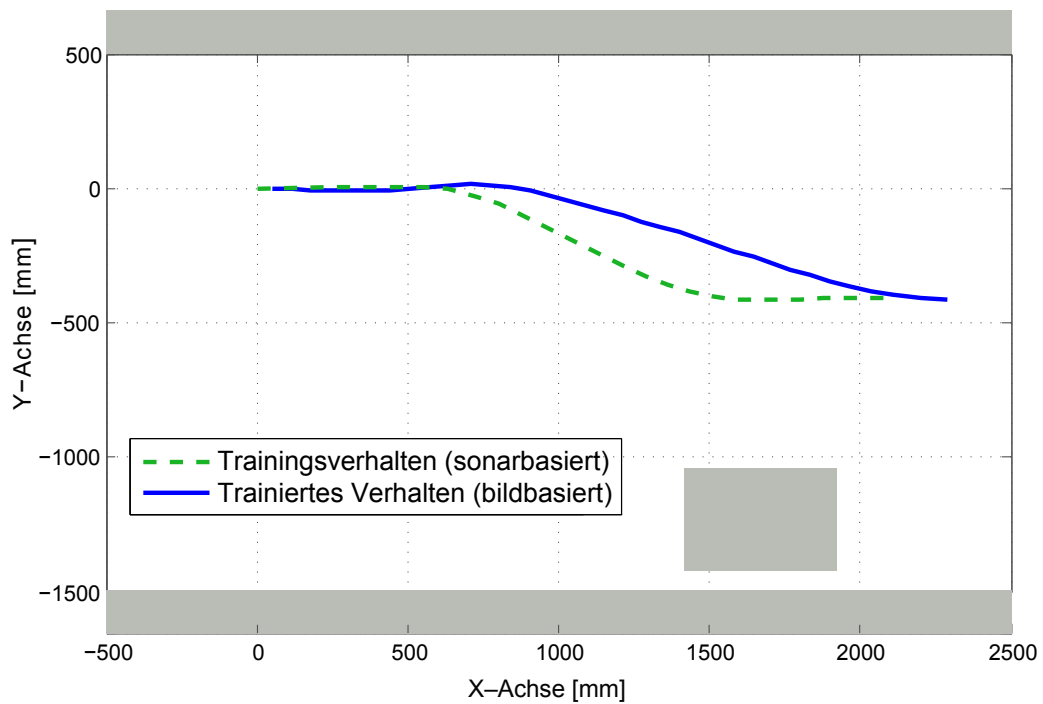
Abbildung 5.10(a) zeigt das Ausweichen des Roboters vor einem Hindernis, das etwas linksseits der Fahrtrichtung des Roboters platziert ist. Die durchgezogene Linie zeigt die Trajektorie des gelernten, bildbasierten Verhaltens, die gestrichelte Linie zeigt die Trajektorie des sonarbasierten Trainingsverhaltens. Startpunkt der beiden Trajektorien ist jeweils links in der Abbildung, das Hindernis ist in der Abbildung oben rechts grau skizziert. Zu sehen ist, dass die Ausweichbewegung beider Verhalten im Prinzip ähnlich ist, die des gelernten Verhaltens jedoch insgesamt träger und weniger stark ist. Dies hängt zum einen mit einer höheren Verzögerung durch die komplexere Wahrnehmung des bildbasierten Verhaltens gegenüber dem sonarbasierten Verhalten zusammen. Zum anderen hängt es damit zusammen, dass das trainierte Verhalten nicht nur auf der aktuellen Situation beruht, sondern eine Mittelung mehrerer gefahrener Ausweichbewegungen unterschiedlicher Umgebungen darstellt.

Abbildung 5.10(b) zeigt eine Fahrt des Roboters in einem Korridor über circa zwei Meter Länge. Startpunkt des Roboters ist links in der Abbildung, der Roboter startet nicht mittig im Korridor, sondern seitlich versetzt. Nach etwa anderthalb Metern befindet sich auf der rechten Seite des Korridors eine Säule, in der Abbildung grau skizziert. Der Umfang der Abbildung gibt die Geometrie des durchfahrenen Korridorabschnitts wider. Erneut ist zu sehen, dass die Fahrt des trainierten Verhaltens (durchgezogene Linie) trägere Lenkbewegungen macht als die des sonarbasierten Trainingsverhaltens. Beide Verhalten zeigen aber eine ähnliche Ausrichtung des Roboters im Korridor mit nahezu übereinstimmender Endposition und -orientierung.

Um die Robustheit des Systems gegenüber wechselnden Umgebungen zu belegen, wurde das System in verschiedenen Innenraum-Umgebungen mit unterschiedlichen Bodenbelägen,



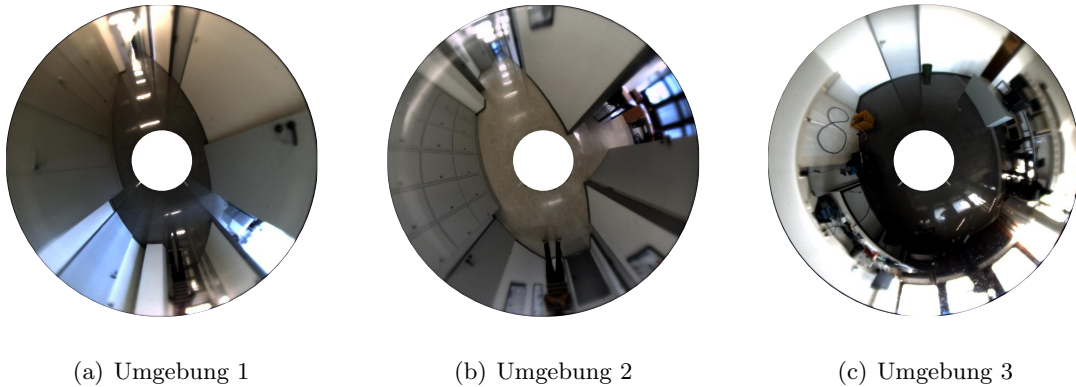
(a) Kollisionsvermeidung



(b) Korridorfolge

Abbildung 5.10: Vergleich der Trajektorie des Trainingsverhaltens und des trainierten Verhaltens in der Draufsicht.

Wandfarben und Gegenständen trainiert und getestet. Abbildung 5.10 zeigt drei der Umgebungen im Vergleich.



(a) Umgebung 1

(b) Umgebung 2

(c) Umgebung 3

Abbildung 5.11: Trainingsumgebungen in der omnidirektionalen Ansicht.

Die Robustheit des Systems kann darüber gezeigt werden, dass ihre Unabhängigkeit von der gewählten Umgebung belegt wird. Dazu wird der Roboter einmal mit den kompletten Trainingsdaten aus allen Umgebungen trainiert und einmal mit allen Trainingsdaten exklusive der Umgebung, in der der Lernfehler bestimmt wird. Werden Trainingsdaten aller drei Umgebungen benutzt, und die Instanzen aus Umgebung 2 zur Validierung benutzt, liegt das erste lokale Minimum des Lernfehlers bei 13 Merkmalen und  $0,033 \frac{\circ}{cm}$ .<sup>1</sup> Werden nur Trainingsdaten der Umgebung 1 und 3 benutzt, und die Instanzen aus Umgebung 2 zur Validierung benutzt, liegt der Lernfehler bei 13 Merkmalen bei  $0,063 \frac{\circ}{cm}$ .

Zu sehen ist, dass der Lernfehler auch in Umgebungen, in denen der Roboter nicht trainiert wurde, niedrig bleibt. Wie zu erwarten steigt der Lernfehler, das Verhalten bleibt aber bei einem Fehler von 6 Grad auf einem einem Meter Strecke funktionsfähig und hat damit die Unabhängigkeit des Systems von den Umgebungsbedingungen – und damit seine Robustheit – in den Grenzen der als Beispiele gewählten Innenraum-Umgebungen gezeigt.

<sup>1</sup>Der von Kapitel 5.2 abweichende Lernfehler kann dadurch erklärt werden, dass eine geringere Anzahl von Trainingsinstanzen benutzt wurde, um die Ergebnisse mit dem folgenden Experiment vergleichbar zu machen. Der Testdatensatz besteht darüber hinaus nicht aus allen Instanzen, sondern nur aus Instnzen aus Umgebung 2.



## Kapitel 6

# Zusammenfassung und Ausblick

Die vorliegende Arbeit zeigt ein Konzept, mit dem grundlegende Roboterverhalten in Innenraum-Szenarien rein bildbasiert und dabei robust gegenüber Veränderungen der Umgebung trainiert werden können. Als Sensorik wurde eine omnidirektionale Kamera in Kombination mit einer monokularen Kamera und einer 3D-Kamera verwendet.

Die Robustheit der Wahrnehmung gegenüber typischen Abweichungen von Innenraumszenarien, wie wechselnde Beleuchtungssituationen und Änderungen in Textur und Farbe der Umgebung, wurde durch Verbesserung der Bildsegmentierung unter Nutzung segmentierter 3D-Informationen erreicht. Dazu wurde die Aufnahme der 3D-Kamera in die drei Klassen „Boden“, „Wand“ und „Hindernis“ segmentiert und anschließend in die monokulare Ansicht übertragen. Dort wurden die segmentierten Bereiche als Starthypothese der Bildsegmentierung genutzt. Darüber hinaus wurde die Bildsegmentierung der omnidirektionalen Ansicht dadurch verbessert, dass verschiedene Bildsegmentierungsverfahren auf Basis der 3D-Informationen in der monokularen Ansicht bewertet werden können. Aufbauend auf dieser Bewertung wird die Segmentierung der Frontansicht in die omnidirektionale Ansicht übertragen, um dem Roboter eine robuste Wahrnehmung seiner lokalen Umgebung zu ermöglichen.

Aus der segmentierten Front- und Umgebungsansicht wurden Merkmale extrahiert, die die Umgebung des mobilen Roboters repräsentieren. Aufbauend auf dieser Repräsentation wurde mittels instanzbasiertem Lernen ein Roboterverhalten trainiert, das Kollisionsvermeidung und Korridorfolge vereint. Das Verhalten wurde nicht getrennt, sondern als ein kombiniertes Verhalten trainiert. Instanzbasiertes Lernen erlaubt dabei durch die lokale Modellbildung die Ausbildung verschiedener Verhaltensmuster innerhalb des trainierten Gesamtsystems. Beim maschinellen Lernen wurden mit der automatischen Merkmalsselektion und dem Clustern der Nachbar-Instanzen typische Probleme des Lernvorgangs adressiert.

Sowohl die Wahrnehmung als auch das trainierte Roboterverhalten wurden auf einem Experimentalsystem in unterschiedlichen Umgebungen validiert. Dabei konnte gezeigt werden, dass sich Fehlzuordnungen bei der Bildsegmentierung (falsch Positive und falsch Negative bezüglich der Bodensegmentierung) durch Nutzung der 3D-Informationen signifikant reduzieren. Das Roboterverhalten konnte erfolgreich die trainierten Verhaltensmuster nachbilden. Damit steht ein robustes Kollisionsvermeidungs- und Korridorfolgeverhalten basierend auf bildgebender Sensorik, und damit die Grundlage für Navigation in Innenraum-Szenarien mit

kostengünstiger und breit einsetzbarer Sensorik zur Verfügung. Die bildgebende Sensorik, die dafür genutzt wird, bietet Potential für viele weitere Anwendungen. Denkbar sind Türdurchfahrtverhalten, Lokalisation und Zielpunktverhalten, die dafür die vorhandene Sensorik nutzen können.

## Ausblick

Im Laufe der Arbeit und unter Berücksichtigung der Ergebnisse hat sich im Bereich der Wahrnehmung und des maschinellen Lernens weiteres Potential gezeigt, das im Folgenden kurz dargestellt wird.

In der vorliegenden Arbeit wird die Semantik des Bildes, die durch die Segmentierung und Klassifizierung der 3D-Ansicht ermittelt wird („Boden“, „Wand“, „Hindernis“), fast ausschließlich genutzt, um die Segmentierung des Bodens zu verbessern. So wird über die Klassifizierung von Bildbereichen eine bessere Anfangshypothese für die Bildsegmentierungsverfahren generiert, die resultierende Segmentierung wird anschließend für die Merkmalsextraktion auf die Bodenfläche reduziert. In dieser Arbeit stehen die segmentierten Bildbereiche „Wand“ und „Hindernis“ einzig in Form ihres prozentualen Anteils in der monokularen Frontansicht als Merkmale für das Roboterverhalten zur Verfügung.

Weiterführende Arbeiten könnten betrachten, inwiefern eine Semantik des Bildes die Robustheit des Verfahrens verbessert, indem nicht nur die Bodenfläche, sondern analog dazu auch Wand- und Hindernisbereiche im Bild durch geometrische Merkmale repräsentiert werden. Dadurch besteht zum Beispiel die Möglichkeit, dass sich unterschiedliche Verhaltensweisen des Roboters bei Ausweichen vor einer Wand oder Ausweichen vor einem Hindernis herausbilden. Auch besteht damit die Möglichkeit, dass das Ausrichten im Korridor nur durch Wände, nicht aber durch Hindernisse beeinflusst wird.

Die Semantik der segmentierten und klassifizierten Frontansicht wird zur Zeit in der omnidirektionalen Ansicht als Starthypothese für die Bildsegmentierung benutzt. Sie findet dadurch nur in einem kleinen Bereich, nämlich dem Schnittbereich der Frontansicht und der omnidirektionalen Ansicht, direkt Anwendung und wird in den Grenzen der Bildsegmentierung in der omnidirektionalen Ansicht propagiert.

Durch *Tracken*<sup>1</sup> einmal segmentierter Bildbereiche in der omnidirektionalen Ansicht könnte die Semantik auch in der omnidirektionalen Ansicht für größere Bildbereiche bekannt gemacht werden. Alle einmalig in der Frontansicht robust segmentierten und gelabelten Bildbereiche können anschließend über den optischen Fluss im Bild oder die Odometrie des Roboters in der omnidirektionalen Ansicht nachverfolgt werden. Dem Tracking kämen in diesem Fall die Distanzinformationen der Bildbereiche, wie sie durch die 3D-Kamera bekannt sind, zugute, da sich die Transformation der Bildbereiche bei Bewegung des Roboters geometrisch exakt bestimmen ließe.

---

<sup>1</sup> *Tracking* (engl., „Spurbildung, Nachführung“) bezeichnet das Verfolgen von Objekten in einer Folge von Aufzeichnungen.



Weiterhin kann die Robustheit der Bildsegmentierung verbessert werden, indem die situationsabhängige Bewertung von Bildsegmentierungsverfahren (siehe Kapitel 3.3) nicht nur zur Wahl der besten *Verfahren* in der aktuellen Umgebungssituation genutzt wird, sondern darüber hinaus zur situationsabhängigen Adaption der *Parameter* der Bildsegmentierungsverfahren genutzt wird. Die in Kapitel 3.3 gezeigte Methodik zur Bewertung von Segmentierungsverfahren kann somit dazu genutzt werden, die Parameter der Bildsegmentierungsverfahren zum Beispiel an wechselnde Umgebungsbedingungen anzupassen.



# Abbildungsverzeichnis

2.1	Das Experimentalsystem mit 3D-Kamera, monokularer Kamera und omnidirektionaler Kamera. . . . .	5
2.2	Schema eines Pixels der <i>PMD[vision] O3</i> . <sup>[2]</sup> . . . . .	7
2.3	(a) Die in dieser Arbeit eingesetzte 3D-Kamera <i>PMD[vision] O3</i> [PMD Tech]	
	(b) Distanzbild dargestellt als Oberfläche. . . . .	8
	(a) 3D-Kamera . . . . .	8
	(b) Distanzbild . . . . .	8
2.4	Katadioptrische Kameraoptik. . . . .	9
	(a) Kameraoptik . . . . .	9
	(b) Umgebungsaufnahme . . . . .	9
2.5	Lichteinfall in einem katadioptrischen System mit Hyperbolspiegel. <sup>[6]</sup> Zu sehen ist, dass das einfallende Licht, das den Brennpunkt $p$ der Kamera passiert, Lichtstrahlen abbildet, die auf den effektiven Blickpunkt $v$ zulaufen. . . . .	10
2.6	Links und Mitte: Konstruktion mit der omnidirektionalen Kamera. Rechts: Durch die Konstruktion eingeschränkte Sichtbereiche der omnidirektionalen Ansicht. . . . .	11
2.7	Unterschied zwischen <i>Lazy Learning</i> und <i>Eager Learning</i> im zweidimensionalen, kontinuierlichen Fall. Mit <i>Lazy Learning</i> kann bei jeder Abfrage das lokale, genauere Modell (gepunktet) gelernt werden, beim <i>Eager Learning</i> nur das globale, ungenauere (gestrichelt). . . . .	16
2.8	Klassifizierung über den Nächste-Nachbarn-Klassifikator. Im Beispiel würde der alleinige Nachbar dem Abfragepunkt $\mathbf{x}_q$ die Problemklasse $A$ zuordnen, die $k$ nächsten Nachbarn (mit $k = 3$ ) klassifizieren den Punkt als $B$ . . . . .	16
3.1	Blickfelder der drei Kameras. Der schraffierte Bereich zeigt den überlappenden Sichtbereich. . . . .	20
3.2	Segmentierung und Klassifizierung einer 3D-Punktwolke: 2D-Ansicht und Boden-, Wand- und Hindernis-Punkte. . . . .	24
	(a) 2D-Ansicht . . . . .	24
	(b) Bodenpunkte . . . . .	24
	(c) Wandpunkte . . . . .	24
	(d) Hindernisse . . . . .	24
3.3	Mit der 3D-Kamera klassifizierte Punkte transformiert in das monokulare Kamerabild als Startpunkte für die Bildsegmentierung. . . . .	27

3.4	<i>Region Growing</i> : Monokulare Ansicht (Links), Bodensegmentierung mit einem Startpunkt (Mitte) und Bodensegmentierung mit Startpunkten aus der 3D-Klassifizierung. . . . .	28
3.5	Watershed-Marker einer Beispielszene. Von links nach rechts: Monokulare Ansicht, Boden-Marker, Wand-Marker und Hindernis-Marker. . . . .	29
3.6	Monokulare Ansicht segmentiert mit <i>Marker-Based Watershed</i> . . . . .	30
3.7	Beziehung zwischen abgerollter und originaler omnidirektionaler Ansicht. . .	33
3.8	Segmentierte omnidirektionale Ansicht der lokalen Umgebung. Das Bodensegment ist optisch hervorgehoben und dient als Grundlage zum Beispiel für die Korridorfolge. . . . .	34
4.1	Krümmung einer Trajektorie in einem Punkt $x$ , gegeben durch die Wegdifferenz $\Delta s$ und die Differenz der Orientierung $\Delta\varphi$ . . . . .	36
4.2	Zu sehen sind die sieben Spalten und jeweils die Höhe des Bodens. In Richtung der Hindernisse ergeben sich niedrige Werte, in Richtung freien Bodens hohe Werte. . . . .	38
4.3	Bodensegmentierung in der omnidirektionalen Ansicht. Aus der Segmentierung kann die Verteilung des Bodens entsprechend der Merkmale ermittelt werden. . . . .	40
4.4	Lernfehler über den Verlauf der automatischen Merkmalsauswahl mittels FSS. . . . .	42
4.5	Roboterfahrt in der Draufsicht: Trainierte Trajektorien $t_0, t_1, \dots, t_4$ und fehlerhaft gemittelte Trajektorie $t^*$ . . . . .	45
5.1	Aufbau des Wahrnehmungsprozesses. . . . .	50
5.2	Vergleich der Bodensegmentierung in einer Beispielszene. Originalansicht (links), Segmentierung mit <i>Region-Growing</i> und einzelner Startpunkt (Mitte) und <i>Marker-based Watershed</i> mit Markern aus der 3D-Klassifizierung. . . . .	51
5.3	Vergleich der Bodensegmentierung in einer Beispielszene. Originalansicht (links), Segmentierung mit <i>Region-Growing</i> und einzelner Startpunkt (Mitte) und <i>Region-Growing</i> mit Startpunkten aus der 3D-Klassifizierung. . . . .	52
5.4	Vergleich der Bodensegmentierung in einer Beispielszene. Originalansicht (links), Segmentierung mit <i>Region-Growing</i> und einzelner Startpunkt (Mitte) und <i>Region-Growing</i> mit Startpunkten aus der 3D-Klassifizierung. . . . .	53
5.5	Schematischer Ablauf des maschinellen Lernens des Roboterhaltens. . . . .	54
5.6	Lernfehler über den Verlauf der automatischen Merkmalsauswahl der ersten 30 Merkmale mittels FSS. . . . .	56
5.7	Vergleich des Lernfehlers der manuellen Merkmalsauswahl und der automatischen Merkmalsauswahl. . . . .	57
5.8	Entwicklung des Lernfehlers bei verschiedenen Lernmodellen. . . . .	59
	(a) Konstantes Modell . . . . .	59
	(b) Lineares Modell . . . . .	59
	(c) Quadratisches Modell . . . . .	59
	(d) Kombiniertes Modell . . . . .	59
5.9	Vergleich der Entwicklung des Lernfehlers mit und Clustering. . . . .	60
5.10	Vergleich der Trajektorie des Trainingsverhaltens und des trainierten Verhaltens in der Draufsicht. . . . .	62
	(a) Kollisionsvermeidung . . . . .	62
	(b) Korridorfolge . . . . .	62

5.11 Trainingsumgebungen in der omnidirektionalen Ansicht. . . . .	63
(a) Umgebung 1 . . . . .	63
(b) Umgebung 2 . . . . .	63
(c) Umgebung 3 . . . . .	63



# Tabellenverzeichnis

3.1	Beispiel der Bewertung zweier Segmentierungsverfahren . . . . .	32
4.1	Ausprägungen der sieben Merkmale am Beispiel. . . . .	38
4.2	Ausprägungen der zwei Merkmale am Beispiel. . . . .	40
5.1	Vergleich der Bildsegmentierungsverfahren bezüglich falsch Positiver. . . . .	52
5.2	Vergleich der Bildsegmentierungsverfahren bezüglich falsch Negativer. . . . .	53
5.3	Übersicht über die Merkmalsauswahl bei verschiedenen Trainingsdaten. . . . .	58





# Literaturverzeichnis

- [1] A. Wilson. Video range camera provides 3-d data. *Vision Systems Design*, 11:19–20, 2006. ISSN 1089-3709.
- [2] B. Schneider. *Der Photomischdetektor zur schnellen 3D-Vermessung für Sicherheitssysteme und zur Informationsübertragung im Automobil*. PhD thesis, Universität-Gesamthochschule Siegen, July 2003.
- [3] I. Schiller, C. Beder, and R. Koch. Calibration of a pmd-camera using a planar calibration pattern together with a multi-camera setup. page 297 ff, 2008.
- [4] S. Baker and S. K. Nayar. A theory of single-viewpoint catadioptric image formation. *International Journal of Computer Vision*, 35(1):1–22, 1999.
- [5] L. Puig, J. Guerrero, and P. Sturm. Matching of omnidirectional and perspective images. 2008.
- [6] G. Kweon, K.T. Kim, Y. Choi, G. Kim, and S. Yang. Catadioptric panoramic lens with a rectilinear projection scheme. *Journal of the Korean Physical Society*, 48:554–563, April 2006.
- [7] C. G. Atkeson, A. W. Moore, and S. Schaal. Locally weighted learning. Technical report, Georgia Institute of Technology, July 1999.
- [8] D. W. Aha, D. Kibler, and M. K. Albert. Instance-based learning algorithms. *Machine Learning*, 6(1):37–66, January 1991.
- [9] Bontempi, Birattari, Bersini, and Bersini. Lazy learning for local modeling and control design. *International Journal of Control*, 72:643–658, 1997.
- [10] G. Bontempi, M. Birattari, and H. Bersini. Lazy learners at work: the lazy learning toolbox. In *In Proceeding of the 7th European Congress on Intelligent Techniques and Soft Computing EUFIT 99*, 1999.
- [11] H. D. Cheng, X. H. Jiang, Y. Sun, and Jing Li Wang. Color image segmentation: Advances and prospects. *Pattern Recognition*, 34:2259–2281, 2001.
- [12] R. Adams and L. Bischof. Seeded region growing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(6):641–647, 1994.
- [13] F. Bonin-Font, A. Ortiz, and G. Oliver. Visual navigation for mobile robots: A survey. *Journal of Intelligent and Robotic Systems*, 53(3):263–296, 2008.

- [14] T. Suwannathat and M. Kaneko. Cooperative usage of monocular camera and omnidirectional camera for segmenting moving humans. In *Machine Vision and Application*, pages 440–443, 2005.
- [15] L. C. Molina, L. Belanche, and A. Nebot. Feature selection algorithms: a survey and experimental evaluation. *IEEE International Conference on Data Mining*, pages 306–313, 2002.
- [16] D. Aha and R. Bankert. Feature selection for case-based classification. *Case-Based Reasoning: Papers from the 1994 Workshop*, 1994.
- [17] T. Athanasiadis, P. Mylonas, Y. Avrithis, and S. Kollias. Semantic image segmentation and object labeling. *IEEE Transactions on Circuits and Systems for Video Technology*, 17(3):298–312, March 2007.
- [18] Benosman R. and S. B. Kang, editors. *Panoramic Vision - Sensors, Theory and Applications*. Springer, 2001.
- [19] M. Betke. Learning and vision algorithms for robot navigation. Technical report, Cambridge, MA, USA, 1995.
- [20] S. Hussmann and T. Liepert. Robot vision system based on a 3d-tof camera. *Instrumentation and Measurement Technology Conference Proceedings, 2007. IMTC 2007. IEEE*, pages 1–5, May 2007.
- [21] M. Lindner, M. Lambers, and A. Kolb. Data fusion and edge-enhanced distance refinement for 2d rgb and 3d range images. *International Journal of Intelligent Systems Technologies and Applications. Issue on Dynamic 3D Imaging*, 5(1):344–354, 2008.
- [22] L.M. Lorigo, R.A. Brooks, and W.E.L. Grimsou. Visually-guided obstacle avoidance in unstructured environments. In *Proceedings of the 1997 IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 1, pages 373–379, Sep 1997.
- [23] R. Reulke. Combination of distance data with high resolution images. In *ISPRS Image Engineering and Vision Metrology*, 2006.
- [24] T. Schamm, J. M. Zöllner, S. Vacek, J. Schröder, and R. Dillmann. Obstacle detection with a photonic mixing device camera in autonomous vehicles. *International Journal of Intelligent Systems Technologies and Applications*, 5(3/4):315–324, 2008.

# Anhang A

## Anhang

### Merkmalskandidaten

Folgende Merkmale standen als Merkmalskandidaten für die automatische Merkmalsauswahl mittels FSS zur Verfügung. Alle Merkmale haben einen Wertebereich von 0 bis 1.

#### Frontansicht

23 Merkmale extrahiert aus der Frontansicht:

- Prozentualer Anteil Boden, Wand und Hindernisse im Bild:  $a_{floor}$ ,  $a_{wall}$ ,  $a_{obstacle}$  (3 Merkmale)
- Höhe des segmentierten Bodenabschnitts in zwölf Spalten:  $h_1, h_2, \dots, h_{12}$  (12 Merkmale)
- Regressionsgerade über  $h_1 - h_{12}$ , eine angenäherte *Bodenlinie*: Y-Achsenabschnitt  $regr_y$  und Anstieg der Geraden  $regr_s$  (2 Merkmale)
- Verhältnis der Menge Boden links und rechts im Bild:  $a_{lr}$  mit

$$a_{lr} = \frac{h_1, h_2, \dots, h_6}{h_1, h_2, \dots, h_{12}} \quad (\text{A.1})$$

(1 Merkmal)

- Verhältnis der Höhe des Bodens jeweils in den gegenüberliegenden Spalten links und rechts im Bild:  $h_{(1,12)}, h_{(2,11)}, \dots, h_{(6,7)}$  mit

$$h_{(1,12)} = \frac{h_1}{h_1 + h_{12}}, \quad h_{(2,11)} = \frac{h_2}{h_2 + h_{11}}, \quad \dots \quad (\text{A.2})$$

(6 Merkmale)

## Omnidirektionale Ansicht

25 Merkmale extrahiert aus der omnidiraktionalen Rundumsicht:

- Prozentualer Anteil Boden im Bild:  $a_{(o, floor)}^1$  (1 Merkmal)
- Höhe des segmentierten Bodenabschnitts in acht Spalten:  $h_{(o,1)}, h_{(o,2)}, \dots, h_{(o,8)}$  (8 Merkmale)
- Menge des segmentierten Bodenabschnitts in acht Spalten:  $a_{(o,1)}, a_{(o,2)}, \dots, a_{(o,8)}$  (8 Merkmale)
- Regressionsgerade über  $h_{(o,1)} - h_{(o,8)}$ : Y-Achsenabschnitt  $regr_{(o,y)}$ , und Anstieg der Geraden  $regr_{(o,s)}$  (2 Merkmale)
- Verhältnis der Menge Boden links und rechts des Roboters:

$$a_{(o,lr)} = \frac{a_{(o,1)}, a_{(o,2)}, a_{(o,3)} a_{(o,4)}}{a_{(o,1)}, a_{(o,2)}, \dots, a_{(o,8)}} \quad (\text{A.3})$$

(1 Merkmal)

- Verhältnis der Menge Boden in Fahrtrichtung und quer zur Fahrtrichtung:

$$a_{(o,corr)} = \frac{a_{(o,1)}, a_{(o,4)}, a_{(o,5)}, a_{(o,8)}}{a_{(o,1)}, a_{(o,2)}, \dots, a_{(o,8)}} \quad (\text{A.4})$$

(1 Merkmal)

- Verhältnis der Höhe des Bodens jeweils in den gegenüberliegenden Spalten links und rechts im Bild:  $h_{(o,1,8)}, h_{(o,2,7)}, h_{(o,4,5)}$  mit

$$h_{(o,1,8)} = \frac{h_{(o,1)}}{h_{(o,1)} + h_{(o,8)}}, \quad h_{(o,2,7)} = \frac{h_{(o,2)}}{h_{(o,2)} + h_{(o,7)}}, \quad \dots \quad (\text{A.5})$$

(4 Merkmale)

---

<sup>1</sup>Der Index  $o$  kennzeichnet Merkmale der omnidirektionalen Ansicht in Abgrenzung zu den aus der Frontansicht extrahierten Merkmalen.

# Anhang B

## Sonstiges

### B.1 Eigenständigkeitserklärung

Hiermit erkläre ich, dass ich die vorliegende Diplomarbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt und Zitate kenntlich gemacht habe.

Dortmund, 10. Juli 2009

Arne Nordmann

## B.2 Aufgabenstellung

Die Forschung am Lehrstuhl RST hat die Entwicklung von Servicerobotern, die ihre Umgebung durch ein Bildverarbeitungssystem erfassen, zum Ziel. Lernen durch Demonstration stellt sich der Herausforderung, Robotersysteme in der gleichen Weise zu trainieren, in der Menschen durch Nachahmung lernen. Im Rahmen dieser Arbeit soll ein bildbasiertes Kollisionsvermeidungs- und Flurfolgeverhalten, basierend auf der visuellen Wahrnehmung der Umgebung durch einen lernbasierten Ansatz, entwickelt werden. Der Roboter wird entweder manuell durch die Umgebung geführt oder navigiert mit Hilfe eines auf Abstandsensoren basierenden Verhaltens. Ein Lernverfahren generalisiert die während der Fahrt aufgezeichneten Bild- und Bewegungsinformation auf ein allgemeingültiges bildbasiertes Navigationsverhalten.

Im Rahmen der Arbeit soll ein bildbasiertes Verhalten zur Korridorfolge und Hindernisvermeidung erlernt werden. Die Bildverarbeitung segmentiert durch Fusion der Bild- und Abstandsinformation die frontale Umgebung anhand von Textur, Farbe und Ebenenzugehörigkeit in die drei Kategorien Flur, Wände und Hindernissen. Die Segmentierung der monokularen Frontansicht wird auf die omnidirektionale Ansicht transferiert. Die Bildinformation wird auf für das Erlernen des Roboterverhaltens relevante Merkmale wie Textur, Farbe und Form der segmentierten Teilregionen reduziert. Mit Hilfe Neuronaler Netze oder instanzbasierter Lernverfahren wird aus den Trainingsdaten auf das allgemeine Verhalten generalisiert. Das Lernverfahren soll durch eine geeignete Repräsentation der Ausgabe in der Lage sein mehrdeutige Situationen und Trainingsdaten zu handhaben. Die Generalisierungsfähigkeit und Robustheit des erlernten Verhaltens soll in Experimenten in unterschiedlichen realen Umgebungen analysiert und verifiziert werden.

Vorgehensweise:

1. Literaturrecherche und Einarbeitung in das Thema
2. Implementierung einer robusten Flur-, Wände- und Hindernissegmentierung mit PMD- und Bildinformationen der monokularen Frontansicht.
3. Generierung einer lokalen Umgebungskarte durch Transfer der segmentierten Frontansicht in die omnidirektionale Ansicht.
4. Definition und Identifikation geometrisch visueller Merkmale basierend auf der Textur, Farbe und Form von Segmenten zum überwachten Lernen bildbasierter Verhalten.
5. Generierung von Trainings- und Validierungsdaten und Erlernen eines bildbasierten Korridorfolge- und Hindernisvermeidungsverhaltens.
6. Evaluierung und Analyse des bildbasierten Verhaltens in Experimenten auf dem mobilen Roboter
7. Dokumentation und Präsentation der Ergebnisse