HAW HAMBURG

BACHELORTHESIS
Eric Denecke

Autonome Nutzung von Aufzügen durch einen Industrieroboter

FAKULTÄT TECHNIK UND INFORMATIK
Department Informatik

Faculty of Computer Science and Engineering
Department Computer Science

HOCHSCHULE FÜR ANGEWANDTE WISSENSCHAFTEN HAMBURG
Hamburg University of Applied Sciences

Eric Denecke

Autonome Nutzung von Aufzügen durch einen Industrieroboter

Bachelorarbeit eingereicht im Rahmen der Bachelorprüfung
im Studiengang *Bachelor of Science Informatik Technischer Systeme*
am Department Informatik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer: Prof. Dr. Stephan Pareigis
Zweitgutachter: Prof. Dr. Henner Gärtner

Eingereicht am: 20. April 2022

Eric Denecke

Thema der Arbeit

Autonome Nutzung von Aufzügen durch einen Industrieroboter

Stichworte

Autonom, Neurales Netz, CNN, Objekterkennung, Klassifizierung, SSD

Kurzzusammenfassung

Der vierrädriger Industrieroboter Husky soll autonom eine Aufzuganlage in einem Gebäude der Hochschule für Angewandte Wissenschaften (HAW) verwenden. Es wird ein fünfteiliger Ablauf erstellt, der die notwendigen Aufgaben eines Stockwerkwechsels abdeckt. Die fünf Phasen werden in Prozesse unterteilt und zu erwartende Probleme analysiert. Die autonome Betätigung der Bedienelemente wird implementiert und getestet. Dafür wird die Lokalisierung der Knöpfe durch Objekterkennung mit dem neuronalen Netz SSD Mobilenet v2 umgesetzt. Das Netz erreicht im Training eine Genauigkeit von 86,4 % mAP. Gedrückt werden die Knöpfe mit dem Greifer des montierten Roboterarms. Für den Kontakt zwischen Greifer und Knopf werden zwei Verfahren entwickelt, die die Bewegung des Arms abbrechen, bevor dieser zu viel Kraft aufwendet. Für die vervollständige Umsetzung des erstellten Ablaufs muss die entwickelte Interaktion mit den Knöpfen um eine autonome Navigation erweitert werden. Die Navigation in Flur und Aufzug wird in der Simulation mit einem erstellten 3D-Modell des Flurs getestet und beobachtete Probleme aufgezeigt.

Eric Denecke

Title of Thesis

Autonomous use of elevators by an industrial robot

Keywords

autonomous, neural network, CNN, object detection, classification, SSD

Abstract

The four-wheeled industrial robot Husky is to use an elevator system in a building of HAW autonomously. A five-part process is created that covers the necessary tasks to change floors. The five phases are divided into processes and expected problems are analyzed. The autonomous operation of the elevator controls is implemented and tested. Object detection with the neural network SSD Mobilenet v2 is implemented to locate the buttons. After training, the network achieves an accuracy of 86.4 % mAP. The buttons are pressed with the gripper of the mounted robotic arm. Two methods are developed to stop the movement of the arm on contact between gripper and button to limit the exerted force. To complete the autonomous process, the developed interaction with the buttons must be extended by an autonomous navigation. The navigation in the corridor and elevator is tested in the simulation within a created 3D model of the corridor and observed problems are pointed out.

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Abkürzungen

BT7 Berliner Tor 7.

HAW Hochschule für Angewandte Wissenschaften.

IMU Inertial Measurement Unit.

mAP mean Average Precision.

RFID Radio Frequency Identification.

ROI Region Of Interest.

ROS Roboter Operating System.

SLAM Simultaneous Localization and Mapping.

SSD Single Shot Detection.

TIQ Testfeld Intelligente Quartiersmobilität.

UGV Unmanned Ground Vehicle.

WLAN Wireless Local Area Network.

Symbolverzeichnis

α Winkel in Grad.

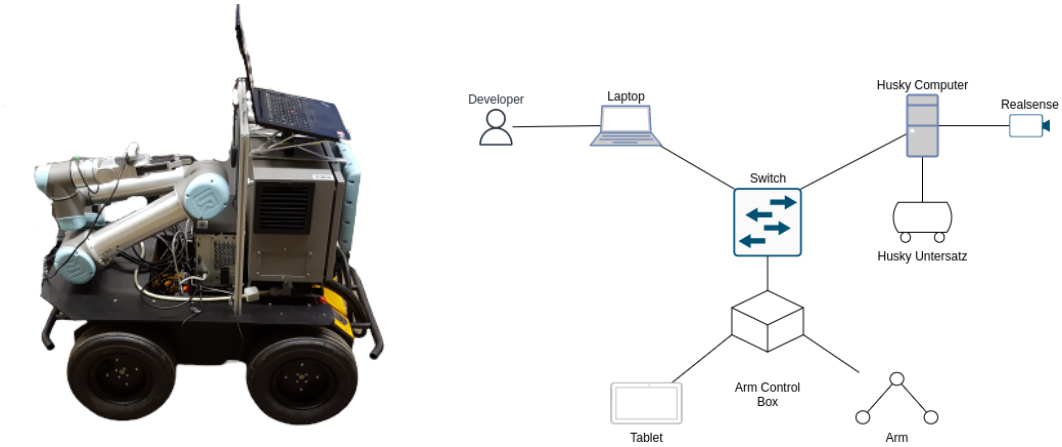
1 Einleitung

Einem vierrädrigen Industrieroboter soll die Navigation in mehrstöckigen Gebäuden ermöglicht werden. Mit einer Bodenfreiheit von 13 cm und einem Achsenabstand von 54 cm ist es dem Roboter nicht möglich, Treppen zu verwenden. Um in mehrstöckigen Gebäuden stockwerkübergreifend agieren können, ist der Roboter daher auf die Verwendung der Aufzuganlage angewiesen.
Ohne spezielle Mechanismen, die eine kontaktlose Interaktion mit der Anlage auf Computerebene erlauben, ist der Roboter auf die manuelle Bedienung angewiesen. Wesentliche Bestandteile sind das Drücken der Knöpfe und das Lesen der Anzeigen. Die Schnittstellen zu Maschinen sind einfach und intuitiv designt, um von Menschen benutzt zu werden. Um mit einem Knopf interagieren zu können, muss dieser vorher identifiziert und geortet werden. Das Erkennen von Knöpfen kann erleichtert werden, indem die Knöpfe mit Markierungen versehen werden, die mit Bildverarbeitungsmethoden gut zu erkennen sind. Dazu zählen Radio Frequency Identification (RFID)-Tags, wie in [1] beschreiben, oder ArUco Marker, die eine Positionierung im dreidimensionalen Raum ermöglichen, sodass diese zum Beispiel von Drohnen zur autonomen Landung verwendet werden können, wie es in [2] eingesetzt wird.
Spezialisierte Roboter, wie sie in [3] vorgestellt werden, sind bereits in der Lage, unterschiedlichste Knöpfe zu erkennen und zu betätigen. Durch ihren experimentellen Aufbau sind sie nicht für den direkten Einsatz auf bereits autonom agierenden Robotern einzusetzen. In dieser Arbeit wird nicht mit einem für die Aufgabe spezialisierter Roboter gearbeitet, sondern mit dem Husky, einem vierrädrigen Unmanned Ground Vehicle (UGV), das mit einem sechsgelenkigen Arm ausgestattet ist. Dieser Arm dient mit seinem Greifer als Interaktionswerkzeug zur Umwelt. Mit ihm sollen nicht nur Objekte wie Türklinken gegriffen, sondern auch Knöpfe gedrückt werden, ohne zusätzliche Hilfsmittel.
Mit Hilfe von Machine Learning werden die Bedienelemente des Aufzugs erkannt und autonom mit dem Arm betätigt. Bei der Objekterkennung wird das neurale Single Shot Detection (SSD) Netz Mobilenet [4] eingesetzt, das für die Inferenz auf eingebetteten Systemen vorgesehen ist.

Ziel ist eine Planung des autonomen Ablaufes, um von einem Stockwerk ins Nächste zu gelangen, unter Verwendung von Aufzügen. Die größte Herausforderung für dieses Szenario ist die Sicherheit gegenüber Personen sowie Objekte, mit denen der Arm physisch interagiert. Der Schutz anderer Fahrgäste im dicht gedrängten Flur oder das sanfte Drücken der Knöpfe durch den kraftvollen Roboterarm müssen zuverlässig unter Kontrolle gebracht werden.
Für die Ablaufplanung werden die Fähigkeiten des Roboters sowie die Räumlichkeiten, in denen er autonom agieren soll, genauer analysiert. Auf diesen Grundlagen wird eine fünfphasige Ablaufplanung erstellt. Diese Phasen werden feiner aufgeschlüsselt und die Erkennung als auch das Drücken von Knöpfen als Prototypen implementiert. Diese Prototypen werden vor Ort an einer Aufzugsanlage getestet. Die Anforderungen an die Navigation werden in der Simulation genauer betrachtet.

2 Der Husky

Um eine Anforderungsanalyse für das geplante Szenario durchzuführen, müssen die technischen Aspekte des Roboters, der die Fahrt durch das Quartier bewältigen soll, als Grundlage festgehalten werden. Zusammen mit der darauffolgenden Betrachtung der Umgebung, in diesem Fall die Aufzuganlage, in der sich der Roboter bewegt, können die Randbedingungen eines erfolgreichen Ablaufes ermittelt werden.
Das betrachtete Quartier, das durch geografische oder soziale Abgrenzung festgelegt wird, ist in dieser Arbeit der Campus der HAW Hamburg. Die Quartiersmobilität, wie sie in [5] beschrieben wird, beschäftigt sich mit dem Transport von Personen oder Gütern durch diesen öffentlichen Raum mit einem Einsatzgebiet von weniger als 3 km.
Der Husky ist ein vierrädriges UGV von der Firma Clearpath Robotics. Der Hersteller [6] beschreibt ihn als eine mittelgroße Entwicklungsplattform, die robust und einfach zu benutzen ist. An der HAW Hamburg wird der Roboter im Rahmen des Testfeld Intelligente Quartiersmobilität (TIQ) für verschiedene Forschungszwecke eingesetzt. Um eine autonome Fahrt im Quartier zu ermöglichen, muss eine Vielzahl an Teilproblemen gelöst werden, darunter die Verwendung von Aufzuganlagen. Hierfür wurde die Ausstattung des Huskys nicht für spezifische Probleme gewählt, sondern ein Roboterarm mit Greifer eingesetzt, dessen Einsatz universal ist. Es gilt mit den gegebenen Möglichkeiten die gestellten Problemfälle zu lösen. Abbildung 2.1a zeigt eine Seitenansicht des Roboters mit allen Aufbauten sowie einem zusätzlich montierten Laptop, der für die Initialisierung und Überwachung von Testabläufen benötigt wird. In Abbildung 2.1b ist die Vernetzung der einzelnen Komponenten aufgezeigt. Der Controller des Arms kann über das angebundene Tablet oder über das Netzwerk gesteuert werden. Im autonomen Betrieb erfolgt die Steuerung ausschließlich über den Husky Computer. An diesen Computer sind die Realsense-Kamera als auch der fahrbare Untersatz angeschlossen. Für die Entwicklung können zusätzliche Geräte in das Netz eingebracht werden, die eine manuelle Steuerung und Überwachung ermöglichen. Die Komponenten sind nach ihrer Aufgabe unterteilt. Im Folgenden werden der fahrbare Untersatz, der verbaute Computer sowie der Roboterarm samt Steuerelementen genauer erläutert. Dazu kommen die Sensoren in Form einer Ka-



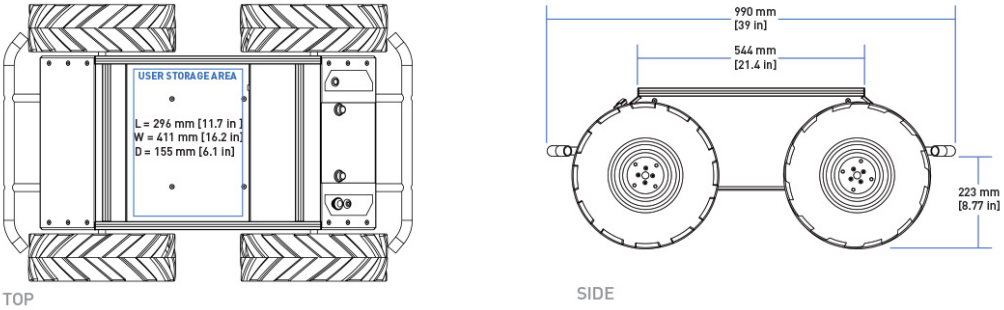
(a) Aufbau während eines Testlaufs (b) Vernetzung der Komponenten in Aufbau a)

Abbildung 2.1: Aufbau des Huskys während eines Testlaufs. In a) ist der Roboter mit seinen Aufbauten und dem Arm in Fahrtposition zu sehen. Rechts ist der graue Kasten der Arm Control Box zu sehen, auf der ein Laptop aufgestellt ist. In b) ist die Vernetzung der Komponenten zu sehen. Laptop, Arm Control Box und der Computer im Husky sind über Ethernet miteinander verbunden. An dem Husky Computer sind Kamera per USB und Untersatz über serielle Schnittstelle angeschlossen. Der Laptop dient zum Starten und Überwachen der Testläufe.

mera und einem Lidar.

2.1 Untersatz

Grundlage des Roboters ist der fahrbare Untersatz, bestehend aus einem Körper, der die elektronischen Komponenten zu Ansteuerung der Reifen beinhaltet, den vier Rädern sowie einem Hohlraum in der Mitte des Körpers, in dem vom Benutzer weitere elektronische Geräte untergebracht werden können. Der Untersatz selbst wiegt ohne Aufbauten 50 kg und kann mit einer Geschwindigkeit bis zu 1.0 $\mathrm{m\,s^{-1}}$ fahren. Die Räder haben einen Durchmesser von 330 mm und besitzen im Gegensatz zur Darstellung in Abbildung 2.2a ein weniger aggressives Reifenprofil, das für Inneneinrichtungen vorgesehen ist.
Die eingebaute versiegelte Blei-Säure-Batterie besitzt eine Kapazität von 480 Wh die bei 24 V 20 Ah liefert. Mit dieser Batterie wird nicht nur das Fahrwerk, sondern auch alle zusätzlich angebrachten Komponenten auf dem Husky betrieben. Für die Geräte sind 5



(a) Draufsicht auf den Untersatz des Huskys ohne Aufbauten

(b) Seitenansicht auf den Untersatz des Huskys ohne Aufbauten

Abbildung 2.2: Technische Zeichnungen des Untersatzes der Fahrplattform Husky. Die Draufsicht in a) beinhaltet die Position und Abmessungen des vorhandenen Stauraums, der in blau dargestellt ist. Die Seitenansicht b) zeigt, dass die Oberkante der Karosserie oberhalb der Räder endet und damit Aufbauten zulässt, die über die Bereifung hinausragen.

V, 12 V, 24 V Anschlüsse vorhanden, als auch ein Spannungswandler auf 230 V mit einer Schutz-Kontakt-Steckdose. Als letztes bietet der Hohlraum einen seriellen Anschluss, über den die Fahrplattform angesteuert werden kann.
Die Breite inklusive der Räder von 670 mm gibt die Mindestbreite von Türen vor, die der Husky durchfahren kann. Aufbauten, die über den Rand der Plattform herausragen, können beim Passieren engerer Stellen, besonders bei Drehungen, zu Kollisionen führen. In Abbildung 2.2b ist zu sehen, dass die Oberkante des Untersatzes oberhalb der Reifen ist. Die in Abbildung 2.1a zu erkennende schwarze Platte, die auf dem Unterbau montiert ist, kann daher die Bereifung überlappen. Sie überragt neben den Reifen auch die angewinkelten Enden der Karosserie und bietet dadurch mehr Platz, um Geräte auf dem Roboter zu montieren. Am vorderen Ende dieser Plattform ist eine Realsense angebracht, die in einer wasserbeständigen Schutzhülle untergebracht ist. Zusätzlich ist mittig auf der Plattform ein Sensorbogen angebracht, der eine erhöhte Montierung von Sensormodulen ermöglicht.
Die Lenkung erfolgt nach dem Prinzip einer Panzersteuerung. Durch eine entgegengesetzte Bewegung der Räder, kann die Fahrtrichtung geändert werden. Obwohl diese Art der Steuerung auch eine Drehung auf dem Punkt zulässt, ist dies nur möglich, wenn einer der Reifen die Haftung verliert. Auf Untergründen mit wenig Haftung ist dies kein Problem, in Innenräumen mit griffigen Bodenbelägen ist zu beobachten, dass das Nachgeben eines der Räder sehr sprunghaft passiert und für eine Vibration der Fahrplattform führt.

2.2 Rechenleistung

Die Ansteuerung des fahrbaren Untersatzes wird über eine serielle Schnittstelle realisiert. Vorgesehen ist, dass der Computer in der User Storage Area (siehe Abb. 2.2a) untergebracht wird. Über die Dauer dieser Arbeit war ein leistungsfähiger Desktop-Rechner verbaut, der aus der Versenkung herausragt. Der Computer ist ausgestattet mit einem Ryzen 5600G als CPU, einer RTX 3060 Grafikkarte sowie mit 16 GB RAM. Als Betriebssystem wird die Linux Distribution Ubuntu 20.04 eingesetzt. Im Gegensatz zu eingebetteten Systemen, wie einem Raspberry Pi oder einem Nvidia Xavier, steht dadurch mehr Rechenleistung zur Verfügung.
Die Wahl der Bauteile wurde nicht anhand der geforderten Leistung unterwegs getroffen, sondern um für weitere Forschungsprojekte zusätzliche Kapazitäten zu besitzen.
Eine Datenverbindung zur Umwelt kann mittels Wireless Local Area Network (WLAN) oder ähnlichen Geräten erfolgen. So ist eine direkte Lenkung des Roboters mittels Controller über Bluetooth möglich.

2.3 Roboterarm

Bei dem Arm handelt es sich um das Modell UR5 der Firma Universal Robots. Der 6-achsige Arm, der in Abbildung 2.3 dargestellt ist, weißt einen maximalen Arbeitsradius von 850 mm auf. Mit seiner Schulter ist der Arm am vorderen Ende der Platte des Huskys befestigt. Um dieses Gelenk kann der Arm gedreht und gehoben werden. Am anderen Ende des Arms befindet sich das Handgelenk, an dem die Werkzeuge befestigt werden können. Mit den drei verbauten Gelenken im Handgelenk kann das montierte Werkzeug um alle drei Achsen rotiert werden. Das sechste Gelenk befindet sich als Ellbogen in der Mitte des Arms.
Der Arm hat eine Traglast von 5 kg und löst bei erhöhtem Kraftaufwand einen Stopp aus, der erst nach einer Bestätigung vom Benutzer weitere Bewegungen des Arms zulässt. Dies ist ein Sicherheitsmechanismus, der bei Kollisionen eine Beschädigung der Umwelt oder der eigenen Hardware vermeidet. Der Arm kann über Netzwerk oder das vorhandene Tablet angesteuert werden. Der Notstopp für den Arm befindet sich ebenfalls auf dem Tablet.
Am Endeffektor des Arms ist ein RG6 2-Finger Gripper von OnRobot montiert, auf dem zusätzliche eine Intel Realsense D435i angebracht ist. Sämtliche Informationen über Arm und Greifer sind nicht nur am Tablet abzulesen, sondern werden im Betrieb über einen

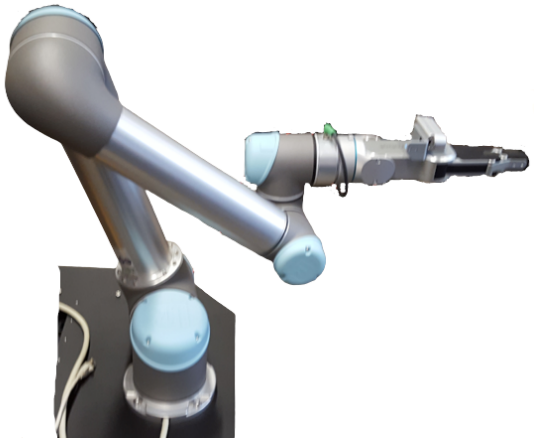


Abbildung 2.3: Roboterarm UR5 der Firma Universal Robots mit montiertem Greifer am Endeffektor. Der Arm ist auf der schwarzen Platte des Huskys montiert. Die zwei Finger des Greifers sind geschlossen, um einen zentrierten Druckpunkt zu bilden.

Computer auch über Roboter Operating System (ROS) veröffentlicht. Darunter sind Informationen wie die Spannungen der Motoren an den einzelnen Gelenken, die Positionen der Gelenke als auch die wirkenden Kräfte und Drehmomente am Endeffektor. Wie in [7] vorgestellt, werden diese Kräfte nicht gemessen, sondern anhand der Spannungen in den Motoren berechnet.
Festgelegte Bewegungen des Arms können mit einer Wiederholgenauigkeit von ±1 mm ausgeführt werden. In Betracht der teilweisen sehr kleinen Knöpfe, die in Aufzügen verwendet werden, ist diese Genauigkeit ausreichend.

2.4 Kamera

An der Front des Untersatzes als auch am Greifer sind je eine Intel Realsense D435i Stereokamera angebracht. Die Kamera verfügt über eine RGB-Kamera, die sich an der linken Seite des Körpers befindet, einem linken und einem rechten Imager, sowie eines Infrarot-Emitters, der für eine bessere Tiefenerkennung bei glatten Oberflächen sorgt.
Die Tiefeninformationen können einzeln abgerufen werden und anschließend manuell mit den Pixeln des Farbbildes verrechnet werden, um eine dreidimensionale Positionserkennung zu ermöglichen. Alternativ kann auf eine Punktwolke zugegriffen werden, die wahlweise sortiert oder unsortiert angeboten wird. Die Auflösung der Farbkamera kann bis zu 1920x1080 Pixel betragen, bei einer Bildrate von 30 Bildern pro Sekunde. Da der

Input neuraler Netze meist kleiner ist, wird die Kamera im Rahmen dieser Arbeit mit einer Auflösung von 640x480 Pixeln und einer Bildwiederholungsrate von 30 Bildern pro Sekunde betrieben. Die im Datenblatt beschriebene Reichweite der Tiefenerkennung liegt bei bis zu 3 m bei voller Auflösung.

2.5 Lidar

Für die Navigation des Fahrzeugs kann ein 3D-Lidar mit 16 Ebenen der Firma Robosense zusammen mit einer XSense MTi-680G RTK Inertial Measurement Unit (IMU) verwendet werden. Dieser kann am Sensorbogen des Huskys montiert werden, sodass er oberhalb der sonstigen Aufbauten positioniert ist. Im Rahmen dieser Arbeit wird der Lidar nur in der Simulation verwendet, da sich der reale Ablauf auf das Erkennen und Drücken der Knöpfe beschränkt, unter der Voraussetzung, dass der Husky sich bereits in Reichweite der Zielobjekte befindet. Der Lidar kann dazu verwendet werden, die Wände im Gebäude mittels Simultaneous Localization and Mapping (SLAM) zu kartografieren. Da es sich um einen Lidar mit wenigen Ebenen handelt und er am höchsten Punkt des Huskys angebracht ist, können flache Objekte/Hindernisse am Boden nicht erkannt werden.

2.6 Roboter Operating System

Als Grundlage für die Kommunikation zwischen den verschiedenen Komponenten wird ROS verwendet. Die einzelnen Komponenten können schrittweise in den Ablauf aufgenommen werden und die Funktionalität der Implementierungen erweitert werden. Da ROS mit einer Publisher-Subscriber-Architektur arbeitet, können der Arm, die Kamera oder der Husky selbst nicht nur vom eingebauten Computer, sondern auch über das Netzwerk angesteuert werden. Eine Auslagerung von rechenintensiven Prozessen, wie etwa die Verwendung von umfangreichen neuralen Netzen, ist möglich. Nachteil dieser Methode ist die Verzögerung, die von der Güte der Verbindung zwischen Roboter und externen Komponenten abhängt.
ROS kann mit den Programmiersprachen C++ oder Python verwendet werden. Mit Blick auf den Einsatz von neuralen Netzen mit dem Tensorflow-Framework wird die Sprache Python gewählt.

3 Umgebungsanalyse

Nachdem der Husky in Kapitel 2 vorgestellt wurde, folgt in diesem Kapitel die genauere Betrachtung der Umwelt, in der sich der Husky zurechtfinden muss. Aufzüge sind weltweit ähnlich gestaltet. Eine allgemeingültige Betrachtung ist durch die vielen unterschiedlichen Einzelheiten aber aufwendig. Daher wird vorerst mit dem Aufbau eines konkreten Gebäudes gestartet. Beispielhaft wird hier das Gebäude der HAW am Berliner Tor 7 verwendet. Die Besonderheit der Anlage ist die Aufteilung der Aufzüge in zwei Gruppen. Eine Gruppe befährt ab dem 2. Stockwerk nur die geraden, die andere Gruppe nur die ungeraden Stockwerke. In den ersten beiden Stockwerken können beide Gruppen mit je drei Aufzügen erreicht werden. Die höheren Stockwerke bieten einen einfacheren Aufbau, in dem nur eine Gruppe mit drei Aufzügen erreicht werden kann. Für den Aufbau wird das 7. Stockwerk gewählt, in dem sich Arbeitsraum mit dem Husky befindet. Das Szenario startet im Flur zu den Aufzügen. Im Folgenden werden dafür der genannte Flur, der Innenraum der Aufzüge sowie die Knöpfe, Schilder und Anzeigen betrachtet, die für einen erfolgreichen Stockwerkwechsel erforderlich sind.

3.1 Der Flur

Startpunkt einer Fahrstuhlfahrt ist der Flur, über den der Aufzug gerufen und betreten wird. Abbildung 3.1 skizziert den Grundriss des Flurs mit den relevanten Maßen. Zusätzlich wird der Husky maßstabsgetreu in verschiedenen Positionen skizziert. Der Flur hat eine Länge von 9 m und eine Breite von 3,3 m. Die Fläche bietet dem Roboter ausreichend Platz zum Manövrieren. In diesem Stockwerk kann der Flur von zwei Seiten durch je eine Tür betreten werden. Der Roboter passt mit seiner Breite von 0,67 m durch die 1,1 m große Öffnung. Bei den Türen handelt es sich um selbstschließende Brandschutztüren, die zu den anliegenden Fluren hin öffnen. Zusammen mit den Aufzugtüren sind diese Bereiche für den restlichen Personenverkehr freizuhalten.

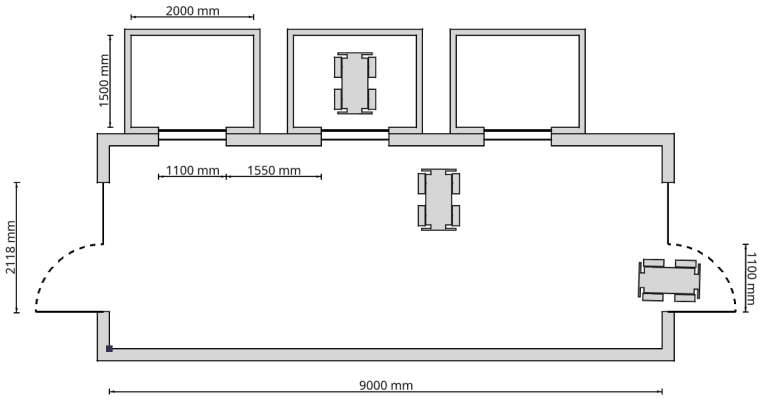


Abbildung 3.1: Skizze des Grundplans des Flurs im 7. Stock des Gebäudes BT7 mit den drei relevanten Aufzügen. Es sind drei beispielhafte Positionen des Huskys eingezeichnet. Die Größe des dargestellten Umrisses des Huskys entspricht dem Maßstab der eingezeichneten Wände. Es ist zu sehen, dass der Roboter im Flur viel Platz hat, in der Aufzugskabine aber wenig Platz für weitere Personen übrig lässt.

Die Türen weisen Fenster zu den benachbarten Fluren auf, es gelangt jedoch kein Tageslicht in beide Räumlichkeiten. Es ist eine gleichmäßige Ausleuchtung vorzufinden, die durch Bewegungssensoren beim Betreten aktiviert wird. Durch das künstliche Licht sind die zu erwartenden Lichtverhältnisse in den meisten Stockwerken gleich.
Zur Orientierung sind an der Wand gegenüber den Aufzügen Plakate und Schilder vorzufinden. Darunter eine Auflistung der Räume in diesem Stockwerk sowie die dort anzutreffenden Personen. Ein Schild mit der Nummer des aktuellen Stockwerks ist dort ebenfalls angebracht. Neben einem Fernseher sind Plakate zu Forschungsprojekten vorzufinden. Am Boden entlang der Wand sind Mülleimer positioniert. Neben diesen Hindernissen sind keine dauerhaften Installationen am Boden vorhanden. Das Abstellen von Kisten oder Geschirr ist untersagt und im Regelfall nicht anzutreffen.

3.2 Die Aufzugkabine

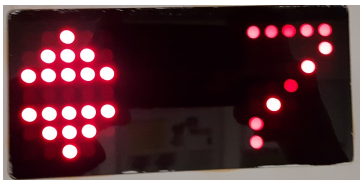
Wie in Abbildung 3.1 dargestellt, sind bis zu drei Aufzüge vom Flur aus erreichbar. Die betrachteten Aufzugkabinen haben eine Grundfläche von 2 m Breite und 1,5 m Tiefe.

Vorgesehen ist der Aufzug für bis zu 18 Personen oder einer Gesamtlast von 1350 kg.
Im Falle der Aufzuganlage in Gebäude BT7 handelt es sich um einen Behinderten gerechten Personenaufzug, weshalb die Bedienelemente in der Kabine zweifach vorhanden sind. Wie in den meisten Aufzügen zu beobachten ist, befindet sich an der Wand neben den Türen ein Bedienfeld, auf dem die Knöpfe vorzufinden sind. Zusätzlich befindet sich auf der rechten Seite eine Konsole mit größeren Knöpfen. Mit einer Höhe von 75 cm ist die Konsole für einen Menschen im Sitzen zu erreichen. Die Gestaltung dieser Bedienelemente ist für den Greifarm des Roboters einfacher zu erreichen und wird deshalb für die Interaktion verwendet.
Um die kleinen Kabinen von Aufzügen größer wirken zu lassen, werden oft Spiegel eingesetzt, die den Raum größer wirken lassen. Im BT7 ist die Kabinenwand gegenüber des Aufzugeingangs mit einem Spiegel versehen, der ab einer Höhe von etwa 90 cm beginnt. Im Gegensatz zu dem Flur ist die Beleuchtung mit einem wärmeren Weiß versehen. Im Zusammenspiel mit den metallenen Wänden ändert sich das Lichtverhältnis im Vergleich zum Flur, ist aber über die verschiedenen Aufzüge konstant.

3.3 Bedienelemente und Schilder

In diesem Abschnitt werden die Knöpfe, Anzeigen und Schilder des BT7 als Fallbeispiel für den Aufbau eines Aufzugs verwendet. Auch wenn Aufzüge im Allgemeinen sehr ähnlich aufgebaut sind, ist die Gestaltung dieser Elemente den Designwünschen des Kunden oder des Herstellers unterlegen.
Als Schnittstellen zum menschlichen Benutzer werden Eingaben durch Knöpfe registriert und Feedback über Display oder lokaler Beleuchtung gegeben. Im Flur sind zwei Knöpfe vorhanden, bis auf im Erdgeschoss und im höchsten Stockwerk, wo der Aufzug nur in eine Richtung gerufen werden kann. Diese dienen dazu, der Aufzuganlage zu signalisieren, dass ein Fahrgast in diesem Stockwerk einen Aufzug benötigt. Die zwei Knöpfe repräsentieren dabei die Richtung, in die der Fahrgast fahren will. Wie auch bei den Knöpfen in der Aufzugskabine, wird ein erfolgreiches Drücken durch das Aufleuchten der Umrandung des Knopfes signalisiert.
In der Kabine ist die Anzahl der Knöpfe höher. Für einen Aufzug, der die ungeraden Stockwerke des Gebäudes bedient, sind neben den einzelnen Stockwerken selbst noch die Tasten zum Öffnen und Schließen der Türen vorhanden, sowie eine Notfalltaste. Die großen Knöpfe auf der Konsole am Rand der Kabine sind quadratisch gestaltet mit einer Seitenlänge von etwa 6 cm. Sie weisen eine schwarze Umrandung und einen weißen

(a) Hoch/Runter Knöpfe im Flur



(b) Display im Flur über den Aufzugtüren mit wartendem Aufzug im 7. Stock

(c) Schild mit der Stockwerkszahl im 6. Stockwerk

Abbildung 3.2: Die drei Arten der Informationen, auf die der Roboter reagieren soll. Die Knöpfe zum Rufen des Aufzugs in a), die Anzeige, in welchem Stockwerk sich der Aufzug befindet in b) und das Schild im Flur mit dem aktuellen Stockwerk in c)

Hintergrund auf, auf dem die Zahlen oder Symbole in Schwarz zu lesen sind. Wird eine Taste betätigt, leuchtet eine orangefarbene Umrandung auf, wie in Abbildung 3.2a zu sehen ist. Die kleineren Knöpfe direkt neben der Tür sind mit einer Fläche von 4 cm mal 3 cm wesentlich kleiner. Die Symbole sind in Schwarz auf silberfarbenem Hintergrund. Die rote Umrandung leuchtet auf, wenn ein Knopf gedrückt wird.
Über den Aufzugtüren befinden sich in der Kabine als auch im Flur ein Display, das das aktuelle Stockwerk anzeigt, in dem sich der Aufzug im Moment befindet (Abb. 3.2b). Die Anzeige wird durch rote LEDs mit einem dunklen Hintergrund realisiert. Zusätzlich zu der Zahl des Stockwerks wird im Flur die Richtung eines eintreffenden Aufzugs angezeigt. Durch einen Pfeil nach oben oder unten wird dem Fahrgast signalisiert, in welche Richtung des eintreffenden Aufzugs weiterfahren wird. Leuchten beide Pfeile, steht ein Aufzug bereit.
An der Wand im Flur gegenüber den Aufzugtüren ist in jedem Stockwerk ein Schild zu finden, dass die Nummer des aktuellen Stockwerks angibt, wie es in Abbildung 3.2c gezeigt wird. Es handelt sich dabei um ein silbernes Schild mit schwarzer Schrift. In den meisten Stockwerken ist darüber hinaus ein Plakat mit der Raumnummern zu finden, die ausweisen, in welcher Richtung die gesuchten Räume zu finden sind.

3.4 Personenverkehr

Während die statischen Hindernisse wie Wände, Türen oder frei stehende Objekte den Husky bei seinen Bewegungen einschränken, besteht das größte Gefahrenpotenzial bei der Interaktion mit anderen Fahrgästen. In dem betrachteten Beispielgebäude variiert das Fahrgastaufkommen stark. Zum Beginn oder Ende einer Vorlesung kann es dazu kommen, dass Flure mit Personen gefüllt sind. Dies ist besonders im 1. und im Erdgeschoss zu beobachten. Um andere Personen nicht zu behindern, werden Türen und Bedienelemente generell freigehalten. Dazu wird beim Eintreffen eines Aufzugs den aussteigenden Personen Vorrang gewährt. Beim Einsteigen in den Aufzug wird der Zugang zu den Knöpfen freigehalten. Während der Fahrt im Aufzug wird der direkte Bereich vor den Türen für früher aussteigende oder neu zusteigende Fahrgäste ebenfalls nicht versperrt. An diese Konventionen sollten sich alle Personen als auch Roboter halten, um Konflikte zu vermeiden.

4 Ablaufplanung

Aus den in Kapitel 2 und 3 vorgestellten Grundlagen können nun konkrete Abläufe für eine erfolgreiche autonome Interaktion mit der Aufzuganlage entwickelt werden. Dabei wird der gesamte Prozess schrittweise zerlegt und analysiert. Wie das Ablaufdiagramm in Abbildung 4.1 zeigt, kann der gesamte Vorgang in fünf Prozesse unterteilt werden. Die Ideen hinter den einzelnen Schritten basieren auf der jetzigen Ausstattung des Huskys, setzen aber Implementationen voraus, die noch nicht vorhanden sind. In den späteren Kapiteln werden einige dieser Prozesse prototypisch implementiert und getestet. Die entwickelten Vorgänge orientieren sich an dem Aufbau der HAW. Der geplante Ablauf beginnt im Flur, von dem aus die Aufzüge betreten werden können. Der Husky muss selbstständig in der Lage sein, sich im Flur zurechtzufinden und die Aufzüge sowie ihre Bedienelemente orten und identifizieren können. Hat er alle nötigen Informationen gesammelt, kann ein

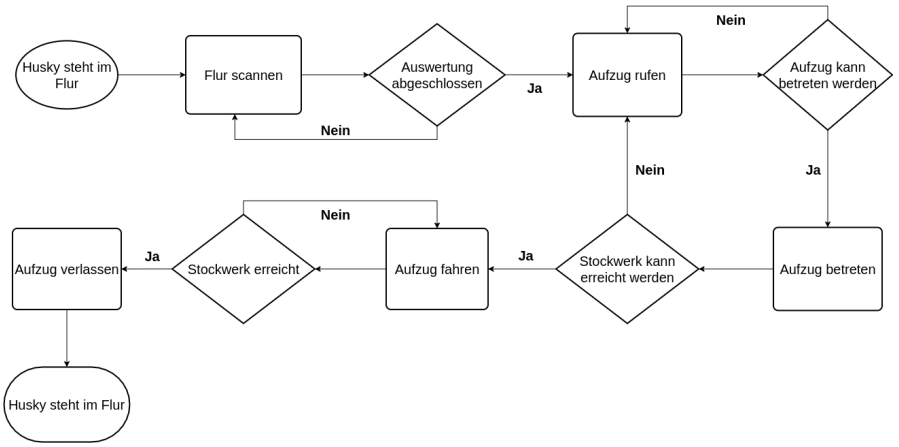


Abbildung 4.1: Ablaufdiagramm der fünf übergeordneten Prozesse, die die notwendigen Schritte für eine erfolgreiche autonome Fahrt mit der Aufzuganlage ermöglichen. Jeder Prozess wird als Phase gesehen, die einen eigenen Ablauf beinhaltet. Eine Phase wird bei anhand eines Kriteriums verlassen. Die Kriterien sind durch die Entscheidungen dargestellt. Start und Endpunkt sind der Flur des Start- und Zielstockwerks.

Aufzug über die entsprechenden Knöpfe gerufen werden. Trifft der gerufene Aufzug ein und es ist ausreichend Platz für den Roboter vorhanden, kann der Husky die Kabine befahren. Nachdem der Husky sein Zielstockwerk gewählt hat, wartet er darauf, dass der Aufzug sein Ziel erreicht. Erkennt der Roboter, dass sein Zielstockwerk erreicht ist, verlässt er diesen und schließt damit das Szenario ab. Die Phasen sind so gewählt, dass bei auftretenden Problemen ein Rückschritt in die letzte Phase möglich ist, um durch erneute Auswertung auf das Problem reagieren zu können.

4.1 Phase 1: Ausgangssituation

Die erste Phase befasst sich mit der Sammlung von Informationen im Flur. Für die Fahrt durchs Quartier soll der Roboter eine Karte besitzen, die für die Orientierung und Navigation verwendet wird. Im Idealfall besitzt der Husky einen digitalen Zwilling des Gebäudes, wodurch eine Navigation verbessert werden könnte, wie es in [8] mit Smart Factorys bereits getestet wurde. Die Prozesse und Entscheidungen, die durchlaufen werden müssen, sind in Abbildung 4.2 dargestellt. Der Start des Szenarios umfasst den Abgleich der aktuellen Daten mit den hinterlegten Informationen in der Karte. Um anderen Personen bei der Bestätigung der Informationen oder dem Warten auf einen Aufzug nicht im Weg zu stehen, ist der Flur in Bereiche unterteilt.
Sind in der Karte keine Zonen hinterlegt, werden diese neu erstellt. Der Zugang zum Flur als auch der Zugang zu den Aufzügen und deren Bedienelemente sind Zonen, in denen der Husky nicht dauerhaft stehen bleiben sollte. Es wird ein Bereich ermittelt, bei dem davon ausgegangen werden kann, dass der Husky niemanden an der Verwendung des Aufzuges hindert und dass er ausreichend Platz zum Bewegen des Arms und der daran befestigten Kamera hat. Im Falle des BT7 kann durch eine Erkennung der Zugangstüren zum Flur und den Aufzugtüren eine Wand im Raum festgelegt werden, an der er nicht im Weg steht. Befindet sich der Husky in der Warteposition, kann die Erfassung der Aufzugelemente beginnen. Wird die Sicht des Huskys durch Personen gestört, kann er eine neue Position im Wartebereich einnehmen, bis er freie Sicht auf den Aufzug hat.
Ein Aufzug selbst ist anhand seiner Türen am einfachsten zu identifizieren. Dies kann mittels Objekterkennung oder durch Messung der tiefer in der Wand liegenden Zugänge erfolgen. Die Erkennung eines Displays über der zugehörigen Aufzugtür bietet eine Vielzahl an zusätzlichen Informationen. Eine intakte Anzeige weist darauf hin, dass der Aufzug betriebsbereit ist. Wie in Kapitel 3 festgestellt, kann die Fahrtrichtung eines eintreffenden Aufzugs erkannt werden. Werden die befahrenen Stockwerke auf dem Display

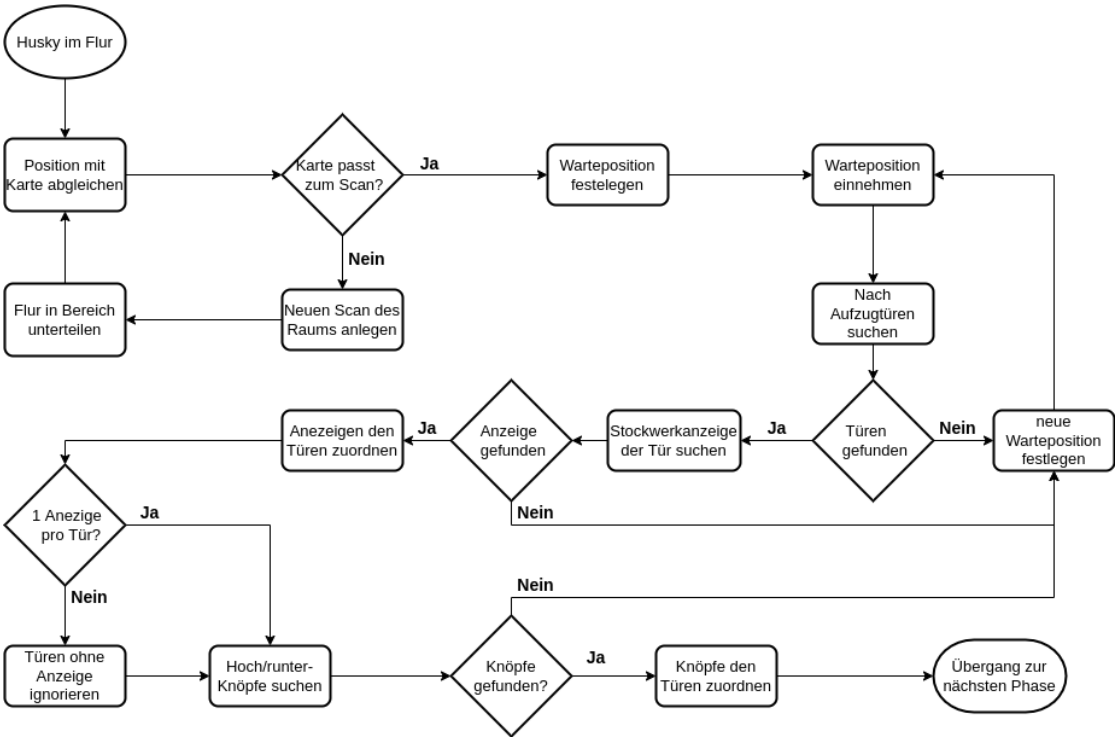


Abbildung 4.2: Ablaufdiagramm der ersten Phase für die Sammlung relevanter Informationen über den Flur und die Aufzuganlage. Ziel ist die Erfassung notwendiger Objekte im Raum sowie die Vermeidung von Behinderung anderer Personen. Kann ein gesuchtes Objekt nicht identifiziert werden, wird der Prozess aus einer neuen Position wiederholt, bis genügend Informationen für mindestens einen Aufzug vorhanden sind.

nachverfolgt, können erste Informationen darüber gesammelt werden, welche Stockwerke mit diesem Aufzug erreichbar sind. Dies ist in dem verwendeten Beispielgebäude von Bedeutung, da der Unterschied zwischen den Aufzügen mit den geraden und ungeraden Stockwerken hieran bereits erkannt werden kann. Als letztes erfolgt die Lokalisierung der Knöpfe zum Rufen des Aufzugs. Die Gruppierung der Türen und Knöpfe zu einer Wand erlaubt die Unterscheidung der beiden Aufzuggruppen mit jeweils drei Aufzügen. Zusätzlich zu den beschriebenen Entscheidungen können Probleme auftreten, die eine erfolgreiche Sammlung der Informationen behindern, diese Problemfelder sind im Folgenden genauer beschrieben.

Problemfeld 01: Blockierte Sicht

Die Sicht auf einige der Elemente der Aufzuganlage können durch Personen blockiert werden. Da der Husky in seiner Warteposition absichtlich Platz für andere Fahrgäste lässt, ist es sehr wahrscheinlich, dass die Sicht der Kamera auf einzelne Elemente unterbrochen wird.
Lösungsansatz: In diesem Fall ist der Roboter darauf angewiesen, die Position der Kamera oder des gesamten Fahrsatzes zu ändern. Dabei ist darauf zu achten, dass der Arm keine unnötig ausladenden Bewegungen vollzieht, um keine Personen zu gefährden und dass er sich möglichst weiterhin im Bereich der Wartezone aufhält.

Problemfeld 02: Wirkung des Roboters auf Menschen

Bei Testläufen der Prototypen in Gebäude BT7 war zu beobachten, dass Studenten, die den Aufzug verwenden wollten, während der Husky im Flur anwesend war, erst zögerten, einen Aufzug zu rufen. Das Einhalten eines gewissen Mindestabstands zu dem Roboter und seinem Arm war zu beobachten.
Lösungsansatz: Durch den Einsatz von Lampen, Projektoren oder Displays auf dem Husky den Personen signalisieren, welche Aktionen als Nächstes von dem Roboter zu erwarten sind.

Problemfeld 03: Aufzug außer Betrieb

Einzelne Aufzüge der Anlage können durch technische Probleme ausfallen. Von Hand erstelle Schilder, die auf einen Betriebsausfall hinweisen, sind für den Roboter nicht interpretierbar. Handelt es sich um den einzigen Aufzug, so muss die gesamte Fahrt unterbrochen werden. Stehen alternative Aufzüge zur Verfügung, wird auf diese ausgewichen.
Lösungsansatz 1: Ist das Display für die Stockwerksanzeige erloschen, kann erst einmal davon ausgegangen werden, dass der Aufzug nicht in Betrieb ist. Sollte er dennoch funktionieren, würde das Erkennen der sich öffnenden Türen für eine normale Weiterfahrt sorgen.
Lösungsansatz 2: Im Zweifelsfall sorgt eine zu lange Wartezeit auf eine Reaktion der Anlage dazu, den aktuellen Vorgang abzubrechen.

Problemfeld 04: False Positives

Abhängig der Qualität der Objekterkennung kann es vorkommen, dass Elemente an Orten erkannt werden, die nicht vorhanden sind.
Lösungsansatz: Da es sich um eine nicht zeitkritische Anwendung handelt, kann durch Mehrfacherkennung bestätigt werden, wenn ein Knopf oder Tür erkannt wurde. Der Suchvorgang kann dabei verlängert oder wiederholt werden.

Problemfeld 05: Objekte außerhalb des Sichtfeldes

Durch das eingeschränkte Sichtfeld der Kamera kann es bei geringer Entfernung zur Wand, an der das zu erkennende Objekt montiert ist, dazu kommen, dass weit oben angebrachte Anzeigen nicht erfasst werden. Als Beispiel kann hier die Anzeige über der Aufzugtür herangezogen werden.
Lösungsansatz: Die Suchvorgänge müssen entsprechend entweder sehr weiträumig oder durch Vorwissen gezielt ausgeführt werden. Eventuell muss der gesamte Roboter dafür seine Position ändern. Da die zu erwartenden Elemente für die Objekterkennung bereits bekannt sind, kann der Suchvorgang bei fehlender Erkennung wiederholt oder ausgeweitet werden.

Problemfeld 06: Informationen speichern

Zum Zeitpunkt dieser Arbeit besitzt der Husky nicht die Kapazitäten mit einer umfangreichen Karte des Gebäudes zu arbeiten.
Lösungsansatz: Erhaltene Informationen können vorerst relativ zur Position des Roboters gespeichert werden, aber ein Festhalten der absoluten Position, sodass sie in der Zukunft wieder verwendet werden kann, ist noch nicht möglich. Die Verwendung einer durch SLAM erstellten Karte oder die Entwicklung eines digitalen Zwillings des Gebäudes würde eine umfangreiche Datenspeicherung zulassen.

4.2 Phase 2: Fahrstuhl rufen

In der ersten Phase wurden alle notwendigen Informationen über die Anlage gesammelt, sodass in dieser Phase der Aufzug gerufen werden kann. Das Ablaufdiagramm in Abbil-

dung 4.3 zeigt die Prozesse, die dabei durchlaufen werden müssen. Die exakte Position des Knopfes gegenüber dem Roboter wird ermittelt. Dabei kann festgestellt werden, ob der Knopf bereits von einer anderen Person gedrückt wurde. In diesem Fall kann direkt in die nächste Phase übergegangen werden. Ist der Knopf außerhalb der Reichweite des Arms, fährt der Husky näher an den Knopf heran. Eine Position senkrecht vor dem zu drückenden Knopf vereinfacht die notwendige Armbewegung zum Drücken. Mit einer Armreichweite von etwa 80 cm, muss der Husky nicht unmittelbar vor dem Knopf stehen, um ihn betätigen zu können. Auch wenn der Roboter parallel zur Wand steht, besitzt der Arm genug Freiheitsgrade, um sich der Wand anzupassen. Dank der drei Gelenke im Handgelenk des Arms kann der Greifer senkrecht zu Wand ausgerichtet werden und auf den Knopf drücken. Nachteilig ist eine ausladende Bewegung des Greifers, die unter Umständen zu Kontakt mit nahestehenden Personen führen kann. Je nach Genauigkeit der Objekterkennung kann die Erkennung in beliebig vielen Schritten während des Drückens wiederholt werden. Der Knopfdruck kann durch haptisches Feedback oder das Aufleuchten der Beleuchtung um den gedrückten Knopf erfolgen. Priorität liegt nicht bei der Geschwindigkeit, sondern bei der Genauigkeit, mit der die Tasten gedrückt werden. Um einen Blick auf die Aufzuganlage zu erhalten, nachdem der Aufzug gerufen wurde, nimmt der Roboter wieder seine Warteposition an der gegenüberliegenden Wand ein. Durch die Entfernung können mehr Elemente im Sichtfeld gehalten werden, was für die nächste Phase relevant ist. Auch in dieser Phase gibt es eine Vielzahl an erwarteten Problemfeldern zu betrachten, die im Folgenden genauer erläutert werden.

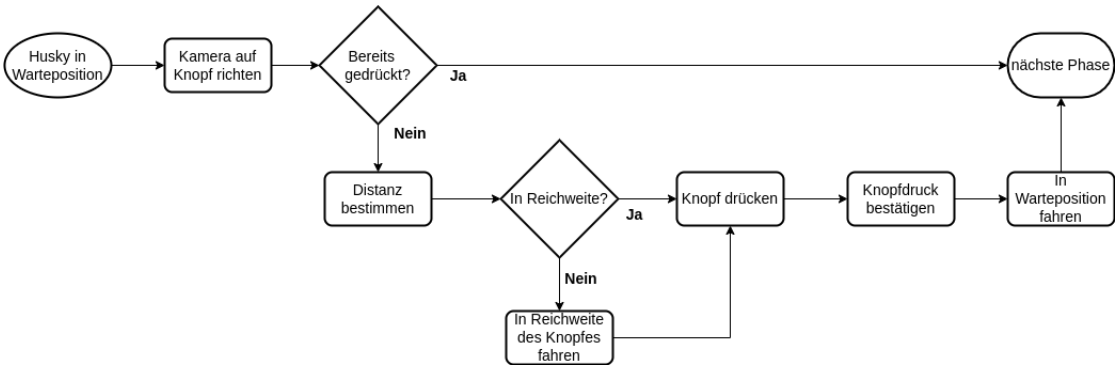


Abbildung 4.3: Ablaufdiagramm der zweiten Phase für die notwendigen Schritte, um einen Aufzug zu rufen. In dieser Phase fährt der Roboter aus seiner Warteposition in Reichweite des Knopfes, um ihn mit dem Roboterarm zu drücken und fährt anschließend wieder in seine Ausgangsposition.

Problemfeld 01: Drücken kleinerer Knöpfe

Bei den betrachteten Knöpfen handelt es sich um Knöpfe mit einer Größe von 6 cm mal 6 cm. Der montierte Greifer besitzt eine Druckfläche von etwa 6,76 cm^2. Für kleinere Knöpfe bei anderen Aufzuganlagen könnte die Druckfläche des Greifers zu groß sein.
Lösungsansatz: Es ist denkbar, dass für die Betätigung von kleineren Knöpfen ein zusätzlicher Aufsatz verwendet wird. Da es sich um einen Greifer handelt, könnte ein Aufsatz mitgeführt werden, der für diese Aufgabe zur Hilfe genommen werden kann. Dieser Aufsatz könnte eine kleinere Druckfläche aufweisen und darüber hinaus noch gefedert gelagert ist, um ein präziseres und sanfteres Drücken zu ermöglichen. Gehalten wird er von den zwei Fingern des Greifers.

Problemfeld 02: Knopfdruck kann nicht bestätigt werden

Ein erfolgreicher Knopfdruck wird durch die dauerhafte Beleuchtung des Knopfes bestätigt. Erlischt die Beleuchtung sofort, deutet es darauf hin, dass ein Aufzug bereits im Stockwerk bereitsteht. Sollte die Hintergrundbeleuchtung ausfallen, kann nicht bestätigt werden, ob der Input angekommen ist.
Lösungsansatz: Im vorliegenden Flur sind die Knöpfe zweifach vorhanden, sodass auf das alternative Bedienfeld ausgewichen werden kann. Ist der Knopf nur ein Mal vorhanden und die Beleuchtung defekt, kann nur das Warten auf das Eintreffen eines Aufzugs aussagen, ob die Interaktion erfolgreich war.

Problemfeld 03: Zu viele Personen im Raum

Während der Stoßzeiten in der Hochschule kann es durchaus vorkommen, dass mehr Personen auf einen Aufzug warten, als mit dem nächsten eintreffenden Aufzug tatsächlich fahren können. Bei solch einer Menge an Menschen im Flur wird der Roboter in seinem Bewegungsfreiraum stark eingeschränkt.
Lösungsansatz: In dieser Situation ist die sicherste Vorgehensweise zu warten, bis ausreichend Platz für den Roboter vorhanden ist.

Problemfeld 04: Knopf wird während des Vorgangs von einer Person gedrückt

Je nach Bewegungsgeschwindigkeit des Roboterarms, kann es vorkommen, dass während des Vorgangs des Drückens der Knopf durch eine Person gedrückt wird. In diesem Fall ist zu vermeiden, dass der Roboter auf die Hand dieser Person drückt.
Lösungsansatz: Wird erkannt, dass ein unerwartetes Objekt zwischen Greifer und Knopf auftaucht, sollte der Vorgang abgebrochen oder verzögert werden. Beim Abbruch sollte detektiert werden, ob der Knopf gedrückt wurde. Im Falle einer Verzögerung wird der Knopf eventuell ein zweites Mal gedrückt, was für die Verwendung der Anlage jedoch keine negativen Folgen hat.

Problemfeld 05: Unerwartetes Eintreffen eines Aufzugs

Sollte fehlerhafterweise nicht erkannt worden sein, dass bereits ein Aufzug gerufen wurde, kann zu jeder Zeit ein Aufzug eintreffen. **Lösungsansatz:** Erfasst der Husky die sich öffnenden Türen mittels Lidar, so kann die Phase zu jeder Zeit abgebrochen und damit begonnen werden, in Phase 3 den Aufzug zu betreten.

Problemfeld 06: Eintreffender Aufzug wird verpasst

Wird der Husky während des Ablaufes durch äußere Einflüsse verzögert, kann es dazu kommen, dass er im Zeitraum vom Drücken des Knopfes bis zur Rückkehr in seine Warteposition den eintreffenden Aufzug verpasst.
Lösungsansatz 1: Der Roboter prüft beim Erreichen der Warteposition den gedrückten Knopf, ob die Beleuchtung immer noch an ist. Ist diese erloschen, aber keine der Aufzugtüren offen, so wurde der Aufzug verpasst.
Lösungsansatz 2: Bereits während der Fahrt prüft der Roboter, ob ein Aufzug eintrifft und geht in die nächste Phase über, bevor er seine Warteposition erreicht.

4.3 Phase 3: Eintreffenden Fahrstuhl identifizieren

Nachdem der Aufzug in der vorherigen Phase erfolgreich gerufen wurde, wird nun darauf gewartet, dass einer der zur Verfügung stehenden Aufzüge eintrifft. Der Ablauf in Abbil-

dung 4.4 zeigt den Start in der Warteposition und die Prozesse und Entscheidungen, die in dieser Phase durchlaufen werden müssen. Ziel ist die Erkennung offener Aufzugtüren und das Befahren der Aufzugskabine. Vom Flur aus sind die einzelnen Zugänge durch versenkte metallfarbene Schiebetüren zu erkennen. Diese Versenkungen sind Merkmale, die zur Identifikation der Zugänge zu den einzelnen Kabinen genutzt werden können. Das sich die Schiebetüren eines Aufzugs öffnen, kann durch eine Veränderung in den Lidar-Daten an den Positionen der Zugänge festgestellt werden. Zusätzlich zeigt das Display über diesen Türen einen roten Pfeil, sobald ein Aufzug eintrifft. Wie in Kapitel 3.4 bereits angesprochen, ist die Wahrscheinlichkeit einer notwendigen Interaktion mit Menschen beim Ein- und Aussteigen sehr hoch. Ist der eingetroffene Aufzug erkannt, muss potenziellen Fahrgästen das Aussteigen ermöglicht werden. Wurden aussteigende Personen durchgelassen, muss zusätzlich auf andere einsteigende Personen geachtet werden. Ist danach noch genug Platz in der Kabine, sodass der Husky hineinfahren kann und genug Platz zum Drücken der Knöpfe hat, kann der Roboter den Aufzug befahren. Durch den kleinen Raum, den die Kabine bietet, können etwa 3 Personen gedrängt gleichzeitig mit dem Husky im Aufzug mitfahren. Diese Personen dürfen nicht zwischen dem Husky und den Bedienelementen im Aufzug stehen. Idealerweise verwendet der Roboter den Aufzug nur, wenn sonst keine Personen anwesend sind. Die Betrachtung der Manövrierbarkeit in der Kabine erfolgt in Kapitel 5. Ist nicht genug Platz in der Kabine, so wird der Vorgang abgebrochen und in die vorherige Phase zurückgegangen und ein neuer Aufzug gerufen. Beim Befahren des Aufzugs können die verfügbaren Knöpfe bereits ein erstes Mal erkannt werden. Dabei ist die Bestimmung der Position nicht entscheidend. Es geht darum, festzustellen, ob das Zielstockwerk mit dem Aufzug tatsächlich erreicht werden kann. Wird beim Betreten festgestellt, dass unter den Knöpfen nicht das gesuchte Stockwerk auswählbar ist, muss der Husky den Aufzug wieder verlassen. Durch den bereits angesprochenen komplexen Umgang mit anderen Fahrgästen, sind dynamisch auftretende Probleme zu erwarten, die in den folgenden Problemfeldern erläutert werden.

Problemfeld 01: Fahrgäste aussteigen lassen

In Aufzügen als auch in Bahnen oder sonstigen Transportmitteln ist es üblich, aussteigenden Personen den Vortritt zu lassen. Dafür muss der Husky entscheiden können, ob Personen aus dem Aufzug aussteigen wollen und wann diese den Aufzug verlassen haben. **Lösungsansatz:** Es wird im angemessenen Abstand eine festgelegte Zeit gewartet. Ist

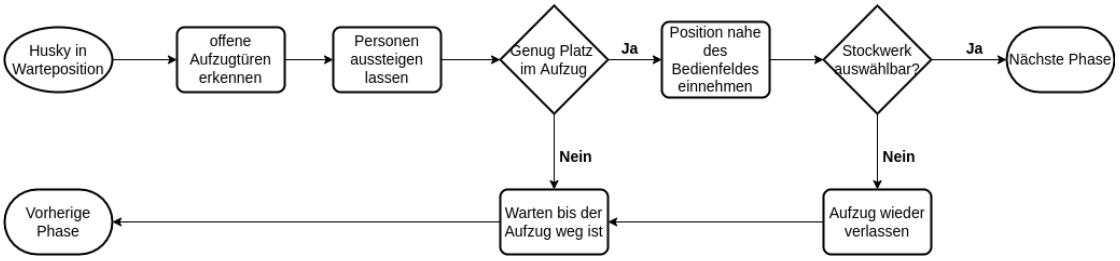


Abbildung 4.4: Ablaufdiagramm der dritten Phase für das Betreten eines eingetroffenen Aufzugs. Sind die offenen Türen eines Aufzugs erkannt, erfordert der Umgang mit anderen Fahrgästen eine Koordination nach üblichen Höflichkeitsregeln. Aussteigende Gäste haben Vorrang und der Aufzug wird nur betreten, wenn ein ausreichender Sicherheitsabstand eingehalten werden kann.

der Zugang dann frei, kann der Husky den Aufzug betreten.

Problemfeld 02: Position in Reichweite der Bedienelemente blockiert

Damit der Roboter den Personen nicht durch Armbewegungen gefährdet, muss ein Sicherheitsabstand eingehalten werden. Dadurch kann es dazu kommen, dass eine einzelne Person, die sich mittig in die Aufzugskabine stellt, keinen Platz für den Husky lässt. In Gebäuden mit viel Personenverkehr kann es dadurch dazu kommen, dass der Husky lange Zeit warten muss, um überhaupt einen Aufzug betreten zu können.
Lösungsansatz: Kann der Husky durch Displays oder Lampen anderen Fahrgästen mitteilen, welche Aktionen als Nächstes ausgeführt werden, kann den Personen im Aufzug signalisiert werden, dass die Bedienelemente angesteuert werden und daher freizuhalten sind.

Problemfeld 03: Personen erkennen

Um einschätzen zu können, ob Personen aus dem Aufzug aussteigen möchten oder weitere Personen im Flur sind, die eventuell in den gleichen Aufzug einsteigen möchten, müssen diese erkannt werden.
Lösungsansatz 1: Mit einem bekannten Grundriss des Raumes können alle sonstig auftretenden Hindernisse als Personen interpretiert werden. Dies erfordert einen Abgleich

der aktuell erzeugten Lidar-Daten und der bereits erstellen Karte.
Lösungsansatz 2: Das Kamerabild kann neben der Objekterkennung auch zur Personenerkennung verwendet werden. Durch ihre hohe Auflösung muss der Schutz der Privatsphäre beim Einsatz berücksichtigt werden.

4.4 Phase 4: Fahrt in der Kabine

Das Ablaufdiagramm in Abbildung 4.5 zeigt, startet die vierte Phase damit, dass der Husky im Aufzug steht. Er ist in Reichweite der Bedienelemente und bereit, sein Zielstockwerk zu wählen. Mit Hilfe der Objekterkennung werden die Knöpfe auf dem Bedienelement exakt lokalisiert. Da die Tasten aus der vorherigen Phase bereits darauf untersucht wurden, dass das angestrebte Stockwerk auch erreichbar ist, kann der Suchlauf nach den Knöpfen gezielt gestartet werden. Ist der Knopf erkannt und seine Position relativ zum Roboter bestimmt, beginnt der Drückvorgang mit dem Arm. Im Gegensatz zur zweiten Phase muss in der engen Kabine besonders darauf geachtet werden, dass der lange Greifer des Arms keine Kollisionen verursacht. Vorteilhaft ist eine Position dicht am Bedienfeld mit einem Mindestabstand von einer Greiferlänge.
Nachdem der Knopf gedrückt wurde und die anhaltende Hintergrundbeleuchtung dies bestätigt, muss der Roboter bestimmen, in welchem Stockwerk er sich über die Fahrt hinweg befindet. Eine Möglichkeit ist es, die Kamera auf den gedrückten Knopf zu richten und die Beleuchtung zu beobachten. Erlischt diese, so ist das Zielstockwerk erreicht. Alternativ kann der Husky seine Kamera auf das Display über der Tür richten und die Stockwerksanzeige dort ablesen, wie es im Ablaufdiagramm 4.5 umgesetzt wurde. Dies hat den Vorteil, dass der Arm nicht permanent auf den Knopf gerichtet ist. Die Implementierung zeigt, dass, um die Beleuchtung eines einzelnen Knopfes zwischen der Vielzahl an anderen Knöpfen zu bestimmen, die Distanz nicht zu groß sein sollte. Die geöffneten Türen können, wie bereits beim Einsteigen angewendet, durch eine Veränderung in der gemessenen Entfernung mit dem Lidar erkannt werden. Zeigt das Display das Zielstockwerk an und die Türen sind geöffnet, kann in die nächste Phase übergegangen werden.
Probleme, die über die zu treffenden Entscheidungen hinaus zu erwarten sind, werden in den folgenden Problemfeldern genauer betrachtet.

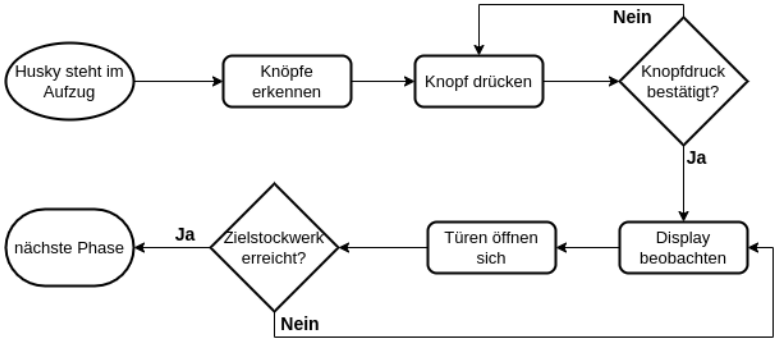


Abbildung 4.5: Ablaufdiagramm der vierten Phase für das Verhalten während der Fahrt im Aufzug. Wie in Phase 2 werden die nötigen Knöpfe erkannt und anschließend gedrückt. Die Information, in welchem Stockwerk sich der Husky befindet, kann er von dem Display über der Tür ablesen.

Problemfeld 01: Fehlerkennungen im Spiegel

Beim Einsatz der Kamera muss darauf geachtet werden, das Aufzüge vermehrt Spiegel in der Kabine einsetzen, um sie größer wirken zu lassen, wie in Kapitel 3 festgestellt. Für den Roboter bergen diese Spiegel die Gefahr, in ihnen erkannte Objekte als echt und nicht als Reflexion anzusehen.
Lösungsansatz: Im Beispiel des Gebäudes BT7 sind die Bedienelemente auf der Konsole am Rand unterhalb des Spiegels montiert. Gefahr einer Spiegelung besteht bei der Beobachtung des Displays über der Tür. Es muss beim Training des neuralen Netzes darauf geachtet werden, dass gespiegelte Zahlen auf dem Display entweder trotz Spiegelung richtig interpretiert oder ignoriert werden.

Problemfeld 02: Zusteigende Fahrgäste

Je mehr Stockwerke der Roboter mit dem Aufzug zurücklegt, desto höher ist die Wahrscheinlichkeit von zusteigenden Fahrgästen. Problematisch an dieser Situation ist, dass der Husky die Bedienelemente blockiert.
Lösungsansatz: Der Raum in der Kabine lässt keinen Platz, um den Roboter vom Bedienfeld weg zu navigieren. Die neu dazu gestiegenen Personen müssen um den Husky herum greifen, um ihr gewünschtes Stockwerk zu wählen.

Problemfeld 03: Ausfall des Aufzugs

Sollte es in dieser Phase zum Ausfall des Aufzugs kommen, so sitzt der Roboter fest.
Lösungsansatz: Eine Interaktion mit der Notfallhilfe ist im Rahmen des Notfallknopfes möglich, aber sonst bleibt dem Roboter wie auch menschlichen Fahrgästen nur die Wahl, auf Hilfe zu warten.

Problemfeld 04: Stockwerk verpasst

Durch zugestiegene Fahrgäste kann der Roboter daran gehindert werden, zu erkennen, dass sein Zielstockwerk erreicht wurde, oder sie blockieren den Weg aus der Kabine hinaus. Der Husky setzt damit seine Fahrt im Aufzug über sein Zielstockwerk hinaus fort.
Lösungsansatz 1: Der Husky verweilt im Aufzug und drückt sein Zielstockwerk erneut. In diesem Fall muss er darauf warten, bis die von anderen Personen gewählten Stockwerke befahren wurden, bevor der Aufzug seine Fahrtrichtung wieder ändert.
Lösungsansatz 2: Der Husky steigt bei der nächsten Gelegenheit aus und beginnt den gesamten Prozess von vorne mit gleichem Zielstockwerk, nun jedoch mit neuem Startstockwerk.

4.5 Phase 5: Aussteigen

Der Ablauf der fünften Phase, der im Detail in Abbildung 4.6 zu sehen ist, beschreibt den Prozess des Verlassens der Aufzugskabine. Erkennt der Husky, dass sein Zielstockwerk erreicht ist, bleibt nur noch die Aufgabe des Aussteigens übrig. Sind weitere Fahrgäste anwesend, können diese den Weg aus dem Aufzug blockieren. Zugestiegene Personen können im Türbereich stehen bleiben, sodass sie dem Roboter den Weg versperren, sobald dieser aussteigen möchte. Ebenso können neue Fahrgäste zusteigen, bevor der Roboter Zeit zum Aussteigen hatte. Diesen Personen muss signalisiert werden, dass der Roboter aussteigen möchte. Wie in den vorherigen Phasen bereits festgestellt wurde, sind dafür zusätzliche Installationen von Lampen oder andern Anzeigegeräten notwendig. Ist der Weg frei, kann der Husky aus dem Aufzug hinausfahren. Hat der Husky den Aufzug erfolgreich verlassen, schließt eine Bestätigung des Ziels anhand der Stockwerksanzeige im Flur die Fahrt ab.

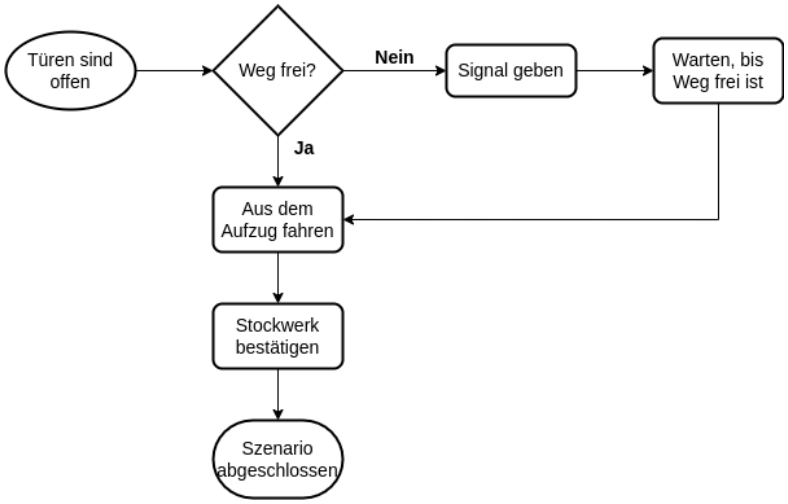


Abbildung 4.6: Ablaufdiagramm der fünften Phase für das Verlassen der Aufzugskabine im Zielstockwerk. Entscheidungen betreffen den Umgang mit Menschen, die den Roboter daran hindern, aus dem Aufzug zu fahren.

Problemfeld 01: Rückwärts fahren

Da der Roboter in der Kabine kaum Platz zum Wenden hat, besonders wenn er nicht allein ist, muss der Roboter rückwärts aus dem Aufzug fahren. Für eine bessere Erkennung mit dem Lidar während einer vorwärts gerichteten Fahrt ist der Lidar um wenige Grad nach vorne geneigt. Die Mindesthöhe, ab der Hindernisse hinter dem Roboter erkannt werden können, ist dadurch höher.
Lösungsansatz: Neben dem Lidar kann auch die Punktwolke der Kamera benutzt werden, um durch die Türöffnung zu navigieren. Dafür muss die Kamera auf dem Greifarm nach hinten gerichtet werden.

5 Autonome Navigation in der Simulation

Bevor mit dem echten Roboter gearbeitet wird, wird eine Simulation des Roboters in Gazebo verwendet. Die Simulation bildet den Husky mit dem montierten Arm sowie dem Sensorbogen mit Lidar nach. Es fehlen der Kasten mit der Steuerhardware des Arms als auch der Computer in der Versenkung des Untersatzes, wie er in Kapitel 2 beschrieben wird. Die Armbewegungen werden trotzdem so eingeschränkt, als wären die Komponenten vorhanden, um entwickelte Abläufe für den realen Arm übernehmen zu können. Als Umgebung wird ein vereinfachtes 3D-Modelle des Flurs und der Aufzüge erstellt, wie es in Abbildung 5.1 dargestellt ist. Die Aufzüge besitzen keine Fahrfunktion und die Schiebetüren lassen sich weder öffnen noch schließen. Zwei der Aufzüge bieten offene Türen, sodass sie vom Husky befahren werden können. Von den Bedienelementen sind die Konsole im mittleren Aufzug und die Knöpfe zum Rufen eines Aufzugs im Flur vorhanden. Sie werden durch einfache Blöcke mit Bildern der entsprechenden Knöpfe als Texturen realisiert.
Da die Objekterkennung der Texturen in der Simulation keine zuverlässige Erkennung bietet und der Arm kein Feedback zu den wirkenden Kräften liefert, können erste Armbewegungen zwar getestet werden, der Kontakt mit echten Knöpfen aber erst in realen Tests umgesetzt werden. Dafür bietet die Simulation ausreichend Spielraum, um die autonome Navigation zu testen. Für das geplante Szenario ist die Navigation durch Flur und Aufzug ein kritischer Abschnitt. Mit den teilweise sehr engen Räumlichkeiten, wie sie in Kapitel 3 beschrieben werden, und der dynamischen Interaktion mit Personen entstehen Situationen, in denen die Bewegungsfreiheit des Roboters stark eingeschränkt ist. Die Navigation wird mittels einer *gmapping* Demo durchgeführt, die als Beispiel von der Firma Clearpath unter [9] frei zur Verfügung gestellt wird. Abbildung 5.2b zeigt den erfolgreich kartografierten Flur des Modells aus Abbildung 5.1. Um die in Pink dargestellten Wände ist ein Sicherheitsabstand in Türkis eingezeichnet. Dieser Bereich wird mit einer roten Umrandung abgeschlossen. Befindet sich der Roboter zum Teil in dieser Zone, so ist es dem verwendeten Planer nicht möglich, weitere Wegplanungen vorzunehmen, da er von einer Kollision mit den Wänden ausgeht.

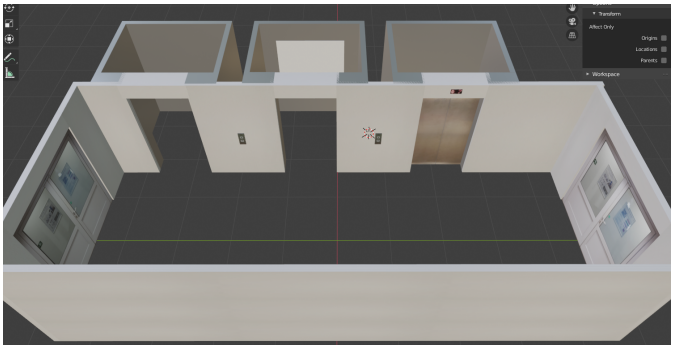
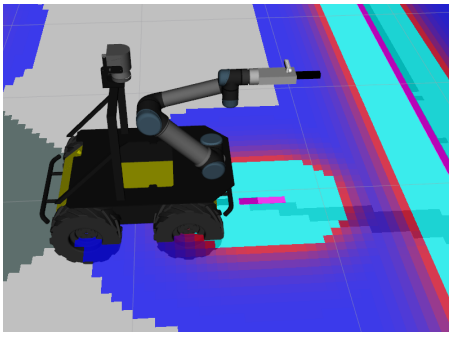
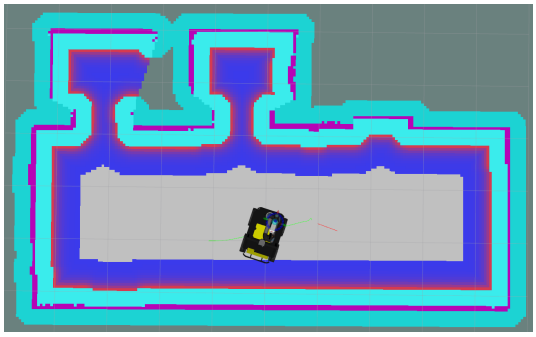


Abbildung 5.1: 3D-Modell des vereinfachten Aufbaus im 7. Stock des BT7. Zwei der drei Aufzüge sind mit offenen Türen dargestellt. Der mittlere Aufzug besitzt die Konsole mit Bedienelementen. Die Knöpfe auf der Konsole als auch die Knöpfe im Flur sind als Texturen vorhanden.

In der erstellten Karte sind die beiden offenen Aufzüge zu erkennen. Der geschlossene Aufzug zeigt einen erkennbaren Versatz in der sonst geraden Wand. Dieses Merkmal könnte zur Identifizierung eines Zugangs genutzt werden. Die erstellte Karte bietet einen *Frame*, in dessen Koordinatensystem erkannte Objekte gespeichert werden können. Die Positionsbestimmung in diesem System ist zuverlässiger als die Verwendung der Odometrie. Die Steuerung des Roboters führt bei Drehungen zwangsläufig zu teils durchdrehenden Reifen, die die Odometrie verfälschen. Muss für die Interaktion mit einem Knopf der Roboter bewegt werden, so ist die relative Position des Knopfes gegenüber dem Husky nach der Fahrt nicht mehr wie erwartet. Da auf dem echten Roboter zum Zeitpunkt dieser Arbeit kein Lidar montiert ist, wird auf eine autonome Navigation während der Testläufe verzichtet.
Um den Lidar nicht in Fahrtrichtung zu blockieren, muss der Arm so flach wie möglich sein. Sollte der Arm so weit gehoben sein, dass er im Sichtfeld des Lidars ist, wird er als Hindernis interpretiert und verhindert weitere Pfadplanungen. Die Abbildung 5.2a zeigt, dass der Greifer in der *costmap* als festes Hindernis direkt vor dem Roboter angesehen wird, sodass die Vorderreifen bereits im kritischen Bereich sind. Wie in Abbildung 5.3b zu sehen ist, wird der Arm so gefaltet, dass der Greifer entgegen der Fahrtrichtung ausgerichtet ist. In dieser Position ragen keine Teile des Arms über den Untersatz des Huskys hinaus, dem Computer in der Versenkung bleibt genügend Platz und das Sichtfeld des Lidars wird nicht beeinträchtigt.
Abbildung 5.3b zeigt eine Situation aus der Simulation, in der zwei Personen im Aufzug stehen. Die Personen werden vereinfacht durch zwei Zylinder dargestellt, die einen Radius von 30 cm und eine Höhe von 180 cm aufweisen. Die Szene wird von der Decke der Auf-



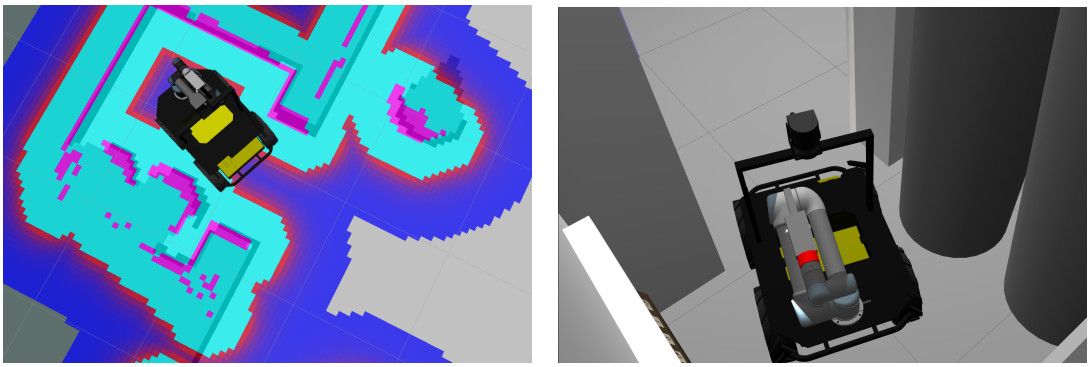
(a) Arm im Sichtfeld des Lidars



(b) Globale *costmap* im 3D-Modell des Flurs

Abbildung 5.2: Vom Lidar erstelle *costmaps*. In a) ist der Arm im Sichtfeld des Lidars und erzeugt ein Hindernis direkt vor dem Husky. Dieses Hindernis blockiert jegliche Wegplanung. In b) ist die *costmap* für das 3D-Modell des Flurs, wenn der Arm in seiner Fahrtposition ist. Ohne Behinderung des Lidars kann erfolgreich eine Karte des gesamten Raums erstellt werden. Offene Aufzüge können eingesehen werden und geschlossene Aufzugtüren durch Tiefenversatz erkannt werden.

zugskabine gezeigt. Die zwei Personen sind rechts im Bild zu sehen. Gegenüberliegend ist die Konsole mit den Knöpfen. Sie wird teilweise durch das in Weiß dargestellte Geländer verdeckt. Der Husky versucht mittels autonomer Navigation in den Aufzug zu fahren. Anhand der *costmap* in Abbildung 5.3a ist zu erkennen, dass dem Roboter nicht viel Abstand zwischen Wänden und Personen bleibt. Es wird dieselbe Situation wie in Abbildung 5.3b dargestellt, es handelt sich aber um eine Draufsicht, die zur 3D-Ansicht rotiert ist. Die Personen sind hier links im Bild durch ihr abgerundetes Profil zu erkennen. Das Resultat ist ein Abbruch der Navigation, ohne dass der Roboter seine Zielposition neben der Konsole erreicht. Für Drehungen des Untersatzes ist nicht mehr ausreichend Platz vorhanden. Für zukünftige Implementation der autonomen Navigation mit dem Husky muss entsprechend darauf geachtet werden, dass eine Pfadfindung auch bei sehr wenig Spielraum möglich bleibt. Dazu könnten weitere Sensoren notwendig sein, die am Rande des Untersatzes genauere Messungen liefern und nicht ausschließlich auf den Daten des weit oben montierten Lidars basieren, die als Ground Truth dienen.



(a) *Costmap* mit der *gmapping* Demo dargestellt in RViz. Der Husky navigiert in den Aufzug, in dem links im Bild zwei Personen stehen.

(b) Aufnahme aus der Gazebo-Simulation. Gleiche Situation wie in 5.3b aus Sicht der Decke im Aufzug.

Abbildung 5.3: Autonome Navigation des Huskys in den Aufzug, in dem bereits zwei Personen stehen. Dargestellt ist die mit *gmapping* Demo (5.3a) erstellte *costmap* und die Sicht aus der Gazebo-Simulation (5.3b). In beiden Abbildungen handelt es sich um dieselbe Situation aus verschiedenen Blickwinkeln. Dem Planer bleibt nicht genug Platz, um den Roboter weiter in den Aufzug zu navigieren.

6 Objekterkennung mit Neuralem Netz

Für die Erkennung der Tasten und Displays wird Objekterkennung mit Machine Learning eingesetzt. Aufbauend auf der Klassifikation von Bildern liefert die Objekterkennung zusätzlich die Position des klassifizierten Objekts im Bild. Visuell wird das Ergebnis mit einem Rahmen um das erkannte Objekt im Bild dargestellt. Alternativ zeigt [10], dass auch Template Matching eingesetzt werden kann, um einen gesuchten Knopf im Bild zu erkennen. Diese Methode leidet stark unter Rauschen im Bild oder im Umgang mit Knöpfen, für die kein Template vorhanden ist und eignet sich daher nicht für den Einsatz in dieser Arbeit.
In [11] wird sich der Aufgabe gestellt, unterschiedlichste Knöpfe zu erkennen. Dazu teilen sie ihren Ablauf in drei Teile. Erst wird eine Region Of Interest (ROI) festgelegt, danach werden Schrift und Zahlen erkannt und schließlich wird der Knopf gedrückt. Das Verfahren setzt nicht nur auf Objekterkennung, sondern nutzt Bildbearbeitungsmethoden wie Kantendetektion, um das relevante Gebiet vorher einzugrenzen. Anschließend werden nicht die Knöpfe selbst, sondern die Beschriftung erkannt. Dies hat den Vorteil, dass dieser Bestandteil über die Vielfalt an Knöpfen konstanter ist als Form oder Farbe. Insgesamt wurde ein F1-Score von 0,985 erreicht und das Verfahren ist robust gegen Winkel und Entfernung der Kamera, Verschmieren des Bildes und Beleuchtungsunterschiede.
Im Gegensatz dazu bieten neuere Lösungen, die mit einem einzelnen Netz arbeiten, Implementationen mit weniger Aufwand. In dieser Arbeit wird das SSD Mobilenet v2 verwendet und mit selbst erstellen Trainingsdaten aus dem Gebäude BT7 trainiert. Während der Einsatz des Netzes dadurch erst nur in diesem Gebäude zuverlässig einsetzbar ist, kann es später mit einem umfangreicheren Trainingssatz erweitert werden. Stellt sich dieser Ansatz als unzureichend aus, können weitere Verfahren aus den vorgestellten Arbeiten hinzugezogen werden. In den folgenden Abschnitten werden das verwendete neurale Netz sowie dessen Training genauer erläutert.

6.1 Single Shot Detection Mobilenet v2

Das neuronale Netz wird auf der verbauten Hardware im Husky eingesetzt. Dazu soll das Netz möglichst effizient sein. Für die Objekterkennung mit neuralen Netzen muss zum einen ein Rahmen um das Objekt festgelegt werden, die Bounding Box, und dazu das Objekt klassifiziert werden. Bislang verwendete Methoden führen diese zwei Schritte getrennt aus und sind rechenaufwendig, sodass sie teilweise mehrere Sekunden pro Bild benötigen.
Es wird das SSD Mobilenet v2 gewählt, da es bei geringer Hardwareleistung für die geplante Anwendung ausreichend schnelle und gute Ergebnisse liefert. Das Netz ist eine Mischung aus den MobileNets, wie sie in [12] vorgestellt werden, und der Single Shot Multibox Detection, die eine Lokalisierung mit geringem Aufwand erzielt, wie es in [13] gezeigt wird. SSD verzichtet auf *Bounding Box Proposal* und den damit verbundenen Extraschritten von Pixel oder Feature Resampling. Bei dem Netz handelt es such um ein Feed-Forward Convolutional Network mit einer festgelegten begrenzten Anzahl von möglichen Bouding Boxes. Das Ergebnis einer Objekterkennung mit diesem Netz ist ein Bild mit Boxen, die das Label und das Vertrauen (*confidenz*) der Klassifizierung zeigen. Wie [4] zeigt, läuft das Netz auch auf einem Nvidia Jetson AGX Xavier mit 19 Bildern pro Sekunde und verbraucht dabei nur 32 MB Speicher benötigt.

6.2 Training

Das neurale Netz wird mittels Transferlearning trainiert. Dabei wird nicht das komplette Netz neu trainiert, sondern ein bereits trainiertes Netz verwendet. Das vorgestellte SSD Mobilenet v2 wurde auf dem Microsoft COCO Datensatz 2017 aus [14] trainiert. Tatsächlich trainiert wird ein *Feature Extractor*, der die Ausgaben des Netzes auf die neu definierten Klassen umsetzt. Die Klassen entsprechen in diesem Fall den Knöpfen mit der jeweiligen Interpretation des Aufdrucks. Es sind insgesamt neun Klassen definiert, die das neurale Netz für Verwendung der Implementierungen in Kapitel 7 klassifizieren muss. Abbildung 6.1 zeigt die von Hand platzierten Boxen, um die zu erkennen Knöpfe in der Kabine des Aufzugs. Zusätzlich sind noch die Klassen der Hoch- und Runter-Knöpfe im Flur vorhanden.
Die Lernrate wird linear innerhalb der ersten 2.000 Schritte auf 0.8 erhöht und fällt über den Verlauf des Trainings wieder ab. Am Ende des Trainings nach 50.000 Schritten ist die

Lernrate 0. Die Anpassung der Lernrate im Verlauf des Trainings verhindert, dass einzelne Trainingsschritte mit schlechten Ergebnissen im späteren Abschnitt eine zu große Änderung an den Gewichten des Netzes verursachen.

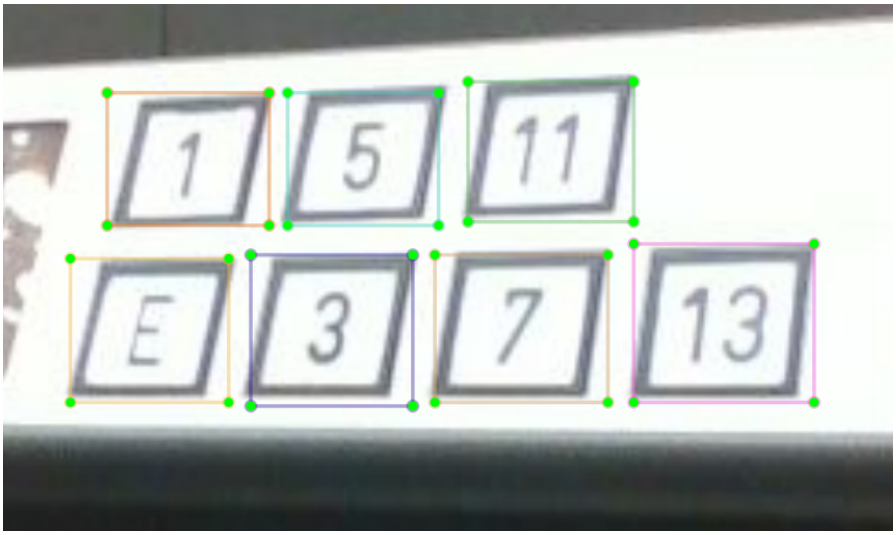


Abbildung 6.1: Beispiel für die manuelle Platzierung der Label-Boxen für die Knöpfe im Aufzug auf der Konsole am rechten Rand der Kabine. Bei einer erfolgreichen Erkennung mit dem neuralen Netz entsprechen die vom Netz gezeichneten Boxen den hier vorgegeben Boxen.

6.2.1 Kamerafeed

Als Eingabe in das Netz wird der Videostream der Realsense auf dem Greifer verwendet. Da das Netz einen Input von 320x320 Pixel verwendet, wird aus dem 640x480 Pixel großem Bild der Kamera ein Ausschnitt gewählt. Der Ausschnitt wird in Abbildung 6.2 dargestellt. Der wegfallende Bildbereich ist dunkler dargestellt als der hellere 320x320 Pixel große Ausschnitt. Der dargestellte Ausschnitt hat den Vorteil, dass die Spitze des Greifers nicht mehr im Bild zu sehen ist, aber trotzdem die Knöpfe auf kurzer Distanz noch teilweise zu sehen sind. Wo der Ausschnitt im Bild positioniert ist, kann je nach Anwendung variiert werden. Zur Aufnahme der Trainingsdaten wird die gezeigt mittige Position gewählt. Beim Einsatz der Kamera ist zu beobachten, dass die automatische Belichtung der Kamera dafür sorgt, dass Bilder für eine Zeit nach der Bewegung stark unter- oder überbelichtet sein können. In den Einstellungen der Kamera kann die automatische Belichtung ausgestellt werden. Wird die Belichtung danach nicht manuelle gesetzt,

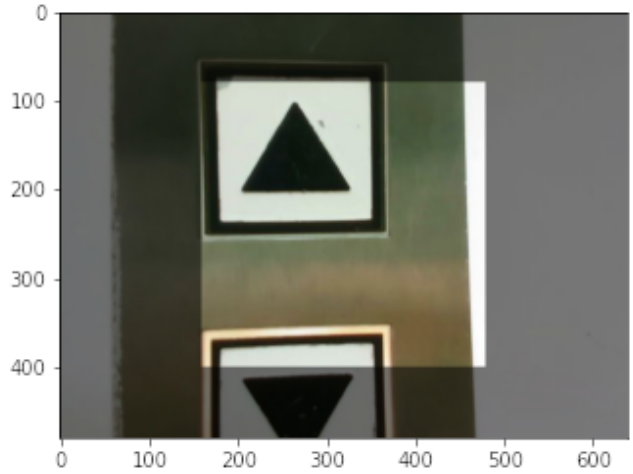


Abbildung 6.2: Aufgenommenes Bild mit der Realsense bei einer Auflösung von 640x480 Pixel. In der Mitte ist der heller dargestellte 320x320 Pixel große Ausschnitt zu sehen, der als Input in das neurale Netz verwendet wird.

bleibt sie auf dem letzten automatisch eingestellten Wert. Da die manuelle Einstellung an die unterschiedlichen Lichtverhältnisse im Flur und der Aufzugskabine angepasst werden müsste, wird vorerst die automatische Belichtung beibehalten. Negative Auswirkungen in der Inferenz werden durch mehr Trainingsdaten bei unterschiedlichen Belichtungen kompensiert.

6.2.2 Vorverarbeitung der Trainingsdaten

Für ein gutes Trainingsergebnis ist die Vielfalt der Bilder, mit denen trainiert wird, entscheidend. Die Vielfalt kann dabei künstlich erhöht werden, indem gemachte Aufnahmen durch verschiedene Vorverarbeitungsschritte um neue erstellte Bilder erweitert werden. Bei den Aufnahmen vor Ort wird darauf geachtet, dass möglichst viele verschiedene Winkel und Entfernungen zu den Knöpfen eingehalten werden. Der Roboter muss die Knöpfe aus seiner Warteposition als auch aus nächster Nähe erkennen können. Dafür wird die Entfernung zu den Objekten während der Aufnahme entsprechend variiert. Da es sich bei den Knöpfen der Aufzüge um beschriftete Objekte handelt, können Vorverarbeitungsverfahren wie das Spiegeln oder übermäßiges Rotieren der Bilder nicht verwendet werden. Die Lesbarkeit von Zahlen oder Buchstaben würde dabei verloren gehen.
Die Kamerabilder der Realsense mit geringer Auflösung weisen von sich aus ein gewisses Maß an Rauschen und Unschärfe auf, besonders wenn die Aufnahme während der Bewe-

gung gemacht wurde. Auf ein zusätzliches Einbringen von Rauschen durch Gauss-Noise oder Motionblur wird daher verzichtet. Durch die in Kapitel 6.2.1 beschriebene automatische Belichtung der Kamera entstehen unterschiedlich helle und dunkle Aufnahmen. Dies wird durch Variation in der Helligkeit und dem Kontrast der Bilder noch weiter ausgeprägt. Da die Knöpfe nicht immer mittig im Bild zu sehen sind, wird eine Translation um bis zu 10 % eingeführt. Zusammen mit einer Rotation mancher Bilder um bis zu 25 % wird die Variation der Knopfpositionen erhöht. Ein Beispiel ist in Abbildung 6.3 dargestellt. Pro Bild wird ein zufälliger Wert innerhalb der gesetzten Grenzen für die einzelnen Anpassungen gewählt. Im Gegensatz zu der Originalaufnahme in Abbildung 6.3a ist das bearbeitete Bild in Abbildung 6.3b gegen den Uhrzeigersinn gedreht und Helligkeit des Bildes wurde reduziert.

(a) original (b) vorverarbeitet

Abbildung 6.3: Vergleich zwischen einer Originalaufnahme a) und einem vorverarbeiteten Bild b). In b) sind die Rotation gegen den Uhrzeigersinn als auch eine Verdunklung zu erkennen. Die Schriftzeichen bleiben lesbar.

6.2.3 Trainingsergebnisse

Wie gut das Training funktioniert hat, kann anhand verschiedener Metriken analysiert werden. Eine schnelle Übersicht bietet eine Betrachtung der Fehlerfunktionen, wie sie in Abbildung 6.4 über den Verlauf des Trainings dargestellt werden. Bei der Objekterkennung wird dabei neben der Regulation zwischen zwei Fehlerfunktionen unterschieden. Der Klassifikationsfehler (*classification loss*) gibt darüber Auskunft, wie groß der Fehler

Tabelle 6.1: Trainingsergebnisse nach 50.000 Schritten mit dem selbst erstellten Trainingssatz und dem Netz SSD Mobilenet v2.

classification loss	localization loss	regularization loss	total loss
3,7468e-4	4,4945e-4	9,0795e-3	9,9036e-3

bei der Bestimmung der Klasse des Objektes ist. Dagegen gibt der Lokalisationsfehler (*localization loss*) aus, wie groß der Fehler bei der Bestimmung der Position des Objektes ist. Der Fehler der Regulation zeigt auf, wie gut der Informationsgehalt auf die verschiedenen Gewichte der Neuronen aufgeteilt ist. Je stärker die Detektion von wenigen Neuronen abhängt, desto größer ist der Regulationsfehler (*regularization loss*). Tabelle 6.1 zeigt die genannten Werte am Ende des Trainings. Die konstant niedrigen Werte der letzten 10.000 Schritte deuten darauf hin, dass das trainierte Netz erfolgreich Objekte detektieren kann. Sie geben allein aber keine Auskunft über die Güte des trainierten Netzes.

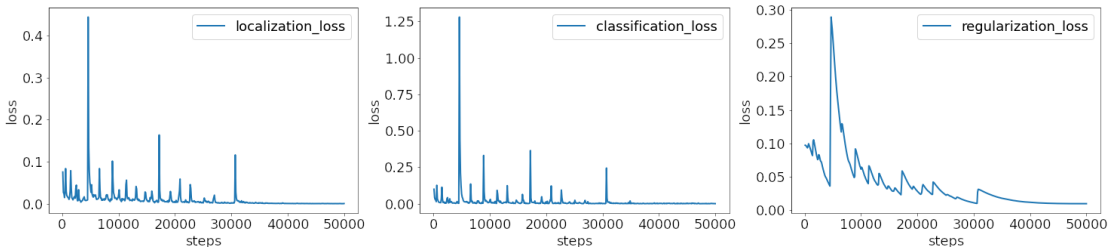


Abbildung 6.4: Verlauf der Fehlerfunktionen über die 50.000 Schritte des Trainings. Zu sehen sind Klassifikationsfehler (links), Lokalisationsfehler(mittig) und Regulationsfehler(rechts). Ab etwa 40.000 Schritte verlaufen die drei Fehlerkurven konstant niedrig. Eine Verbesserung durch ein längeres Training ist nicht zu erwarten.

Neben den Ergebnissen der Fehlerfunktionen können *precision* und *recall* betrachtet werden. *Precision* ist die Prozentzahl an Vorhersagen, die korrekt sind. *Recall* dagegen beschreibt, wie viele der vorhandenen Erkennungen in Relation zu allen möglichen Erkennungen erreicht wurden. Die Güte eines Netzes wird nach der COCO Metrik in mean Average Precision (mAP) ausgedrückt. Wie [15] beschreibt, berechnet sich mAP aus der Relation von *precision* und *recall* unter Berücksichtigung, wie gut die Überschneidung zwischen erkannter Box und tatsächlicher Box ist. Bei dem erstellten Test-Datensatz mit 67 Bildern erreicht das trainierte Netz 86.4 mAP. Im Vergleich zu den mAP von 20.2

in [4], die bei einer wesentlich höheren Anzahl von Objektklassen erreicht wurde, konnte das Netz bei den vorhandenen 9 Objektklassen ein gutes Ergebnis erzielt.

6.2.4 Inferenz

Das neurale Netz liefert pro Durchlauf bis zu 100 Erkennungen, die mit der Wahrscheinlichkeit versehen sind, dass es sich um eine richtige Detektion handelt. Davon werden alle Ergebnisse mit einer Wahrscheinlichkeit über 80 % während des Betriebs veröffentlicht. Ein Ergebnis besteht aus dem Namen des erkannten Knopfes sowie die drei Koordinaten seiner Position im Raum. Von den 30 gelieferten Bildern pro Sekunde werden so viele Bilder bearbeitet, wie die Hardware zeitlich in der Lage ist, zu bearbeiten. Die verwendete Hardware ist in der Lage, die Objekterkennung auf alle 30 Bilder pro Sekunde anzuwenden.
Abbildung 6.5 zeigt den erfolgreichen Einsatz des neuralen Netzes. Es sind die sieben Knöpfe mit ihrer Bounding Box, dem Label und der prozentualen Einschätzung zu sehen. Die Label überschneiden sich und sind in der Abbildung nicht vollständig zu lesen. Es werden alle Knöpfe korrekt und mit einem Vertrauen (*confidenz*) von über 90 % erkannt. In der Inferenz ist das Ergebnis stark von einzelnen Fehlern geprägt. Während

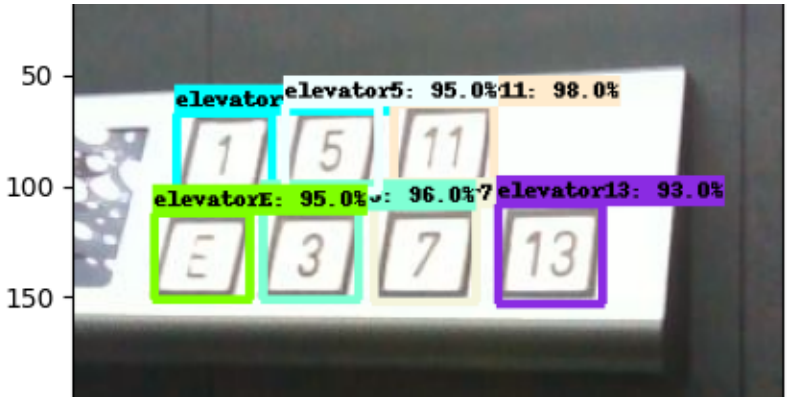


Abbildung 6.5: Ergebnis einer Objektdetektion für die Knöpfe auf der Konsole in der Kabine eines Aufzugs. Es handelt sich um den gleichen Bildausschnitt wie in Abbildung 6.1. Alle Knöpfe werden korrekt erkannt mit einem Vertrauen über 90 %.

die Erkennung der tatsächlichen Knöpfe auf den meisten Distanzen und auch aus großen Winkeln heraus funktioniert, treten immer wieder Situationen auf, in der eine Erkennung

fehlschlägt. Obwohl der Knopf für einen menschlichen Beobachter klar und deutlich im Bild zu erkennen ist, liefert das Netz keine eindeutige Erkennung. Minimale Bewegungen der Kamera können dieses Problem in den meisten Fällen umgehen. Im Verlaufe der Sammlung von Trainingsdaten hat sich erwiesen, dass eine erhöhte Menge von Trainingsdaten dazu führt, dass das beschriebene Problem seltener Auftritt. Als Konsequenz muss beim späteren Einsatz des trainierten Netzes darauf geachtet werden, dass bei zu langer Nichterkennung von Knöpfen eine Bewegung der Kamera für eine erfolgreiche Erkennung sorgen kann.

7 Autonomes Tastendrücken

In diesem Kapitel wird eine Implementierung des Tastendrückens vorgestellt. Es wird vorausgesetzt, dass sich der Roboter mit seinem Arm in Reichweite des Knopfes befindet, der gedrückt werden soll. Der Vorgang kann in einzelne Schritte unterteilt werden, wie sie in Abbildung 7.1 gezeigt werden. Aus seiner kompakten Fahrtposition wird der Arm in eine gezielte Startposition gebracht, von der aus die Kamera am Greifer den gesuchten Knopf anvisieren kann. Nach Erkennung des Objekts und Bestimmung der Position wird der Greifarm bis auf 10 cm an den Knopf herangeführt. Eine erneute Detektion des Knopfes ermöglicht eine genauere Positionsbestimmung. Mit einer zweiten Bewegung direkt auf den Knopf zu wird dieser betätigt. Sobald erfolgreich Kontakt mit dem Knopf hergestellt wurde, kann der Arm über die Startposition wieder in seine Fahrposition überführt werden.
In dem Ablauf werden vier Techniken wiederholt eingesetzt. Diese werden in den fol-

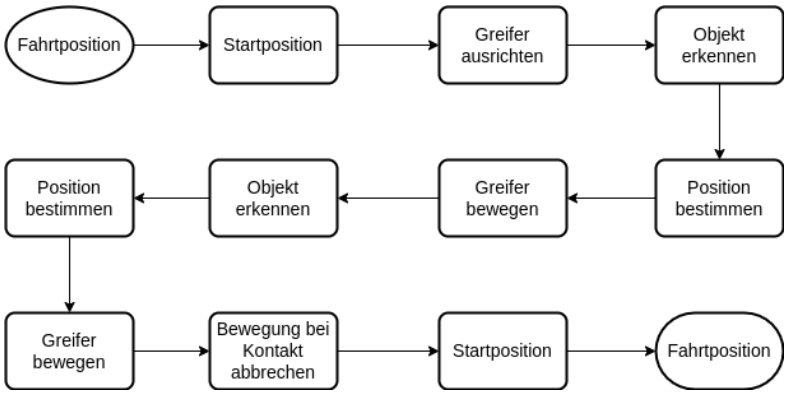


Abbildung 7.1: Ablauf der Bewegung des Arms während eines Tastendrucks. Aus der definierten Startposition heraus wird das Objekt erkannt und seine Position bestimmt. Der Arm wird auf Kurze Distanz vor das Objekt bewegt. Nach einer erneuten Erkennung wird Kontakt mit dem Objekt hergestellt. Bei Kontakt wird die Bewegung abgebrochen und der Arm wieder in Fahrtposition gebracht.

genden Abschnitten genauer beschrieben. Dazu wird die Berechnung der Position eines Objektes im Bild anhand der Kameradaten gezeigt. Ist die Position bekannt, muss der Greifer auf diese Position zubewegt werden. Bei den Bewegungen des Greifers mit der darauf montierten Kamera wird zwischen konkreten Ausgangspositionen und der autonomen Bewegung auf den Knopf zu unterschieden. Es wird gezeigt, wie der Greifer dank des gelenkigen Arms so ausgerichtet werden kann, dass der Drückvorgang unabhängig von der Position des Roboters senkrecht zum Knopf ausgeführt werden kann. Schließlich werden zwei Verfahren untersucht, wie die Bewegung unterbrochen werden kann, sobald der Knopf gedrückt wurde.

7.1 Berechnung der Objektposition

Um die Position von erkannten Objekten im dreidimensionalen Raum zu bestimmen, werden Bild- und Tiefendaten der Kamera zusammengeführt. Von den Eckpunkten der Bounding Box, die mit dem neuralen Netz definiert wurde, wird ein Mittelpunkt des Knopfes in der Bildebene berechnet. Das Ergebnis ist eine Position als Koordinate in x- und y-Richtung in der Bildebene, wie sie in Abbildung 7.2 in grau gezeigt ist. Zu dem Farbbild wird ein Tiefenbild geliefert, dass die Entfernungen zum jeweiligen Bildpunkt enthält. Aus dem Tiefenbild kann die Entfernung $depth_z$ zu dem Punkt im Bild gelesen werden. Zusammen mit den intrinsischen Parametern der Kamera können die tatsächlichen Abstände in X- und Y-Richtung zwischen der optischen Achse und dem gesuchten Punkt berechnet werden. Der gesuchte Punkt wird in der Abbildung 7.2 als P(X,Y,Z) dargestellt. Die Berechnung kann getrennt für die beiden Richtungen erfolgen. Wie in den Formeln 7.1 und 7.2 beschrieben, gehen neben den Pixelkoordinaten $pixel_x$ und $pixel_y$ auch die Position des Hauptpunktes (*principal point*) pp_x und pp_y sowie der Brennweite (*focal lenght*) f_x und f_y ein, um eine Entfernung des Objektes zum Mittelpunkt des Bildes in Zentimeter zu erhalten. Abbildung 7.2 zeigt den Zusammenhang der genannten Parameter, wenn die Kamera im optical center positioniert ist und das erkannte Objekt sich an Punkt P befindet.

$$dist_x = \frac{pixel_x - pp_x}{f_x} \cdot depth_z \tag{7.1}$$

$$dist_y = \frac{pixel_y - pp_y}{f_y} \cdot depth_z \tag{7.2}$$

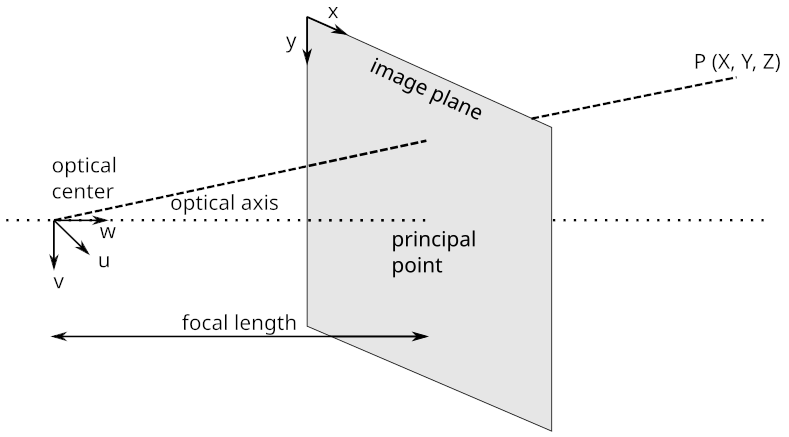


Abbildung 7.2: Kameraparameter *principal point pp* und *focal length f* in Abhängigkeit des *optical center* und der *image plane*. Der Punkt P stellt einen Punkt im Raum dar, der auf dem Bild zu sehen ist. In der *image plane* liegen die Pixel $pixel_x$ und $pixel_y$. Mit der Tiefe P(Z) ($depth_z$) und den dargestellten Parametern können P(X, Y) berechnet werden.

7.2 Armsteuerung

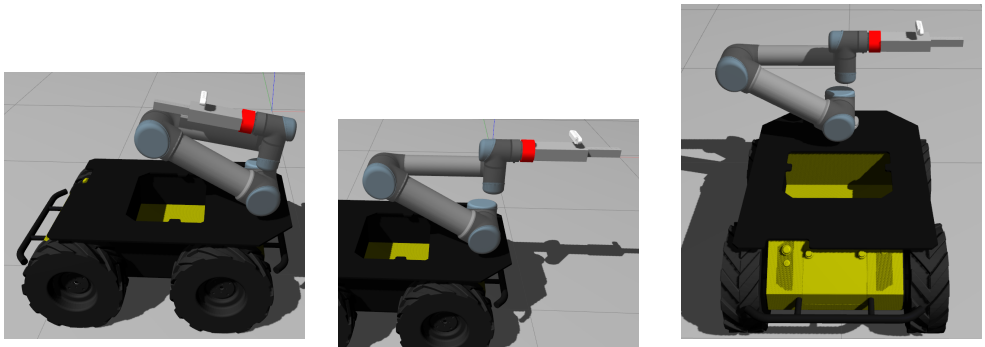
Für die Umsetzung einer Bewegung mit dem Roboterarm stehen drei verschiedene Methoden zur Verfügung. Die einzelnen Winkel der Gelenke können direkt gewählt werden. Diese Methode bietet dem Programmierer die exakte Kontrolle über die Bewegung und Ausrichtung des Arms. Für wiederholte Bewegungen, die den Arm aus einer bekannten Position eine gezielte neue Position lenken, eignet sich dieses Vorgehen am besten, da Start-, Zwischen- und Endpositionen als feste Werte der Winkel vorgegeben werden. Wird nur eine Position des Endeffektors vorgegeben, sodass die Winkel für die Gelenke erst durch inverse Kinematik berechnet werden müssen, verursacht diese Methode viel Aufwand.
Die zweite Methode gibt eine Position des Endeffektors vor und überlässt dem implementierten Planer die Festlegung, welche Positionen die Gelenke einnehmen müssen, damit der Endeffektor an den gewünschten Ort gelangt. Der Weg, den der Endeffektor dabei zurücklegt, ist nicht vorgegeben. Es treten in der Praxis Bewegungen des Arms auf, die den Bewegungsradius ungewollt erhöhen. Dazu zählen eine Rotation des Endeffektors, sodass das Handgelenk oberhalb des Greifers ist oder weitreichende Drehbewegungen um Schulter und Ellbogen des Arms.
Als dritte Methode kann der Arm durch Vorgabe von Wegpunkten linear auf sein Ziel zubewegt werden. Aus den Wegpunkten wird ein Plan erstellt, der gleichmäßige Bewegung

des Endeffektors erzielt. In dieser Bewegung bleibt die Ausrichtung des Endeffektors konstant. Dies ist besonders beim Drücken von Knöpfen vom Vorteil, da eine kontrollierte senkrechte Bewegung auf den Knopf zu erreicht werden kann. Darüber hinaus ist mit der Ausrichtung des Greifers auch gleichzeitig die Ausrichtung der Kamera bekannt.
Für alle drei Varianten der Steuerung ist es möglich, die Geschwindigkeit als auch die Beschleunigung festzulegen. Diese werden dabei als Faktor der maximalen Geschwindigkeit bzw. Beschleunigung gesetzt. Für verwendete Standardpositionen im Ablauf des Tastendrückens werden die Gelenke des Arms einzeln angesteuert. Für die Bewegungen auf ein erkanntes Objekt zu wird die kartesische Wegplanung verwendet.

7.3 Arm Startpositionen

Wie in Kapitel 5 beschrieben, befindet sich der Arm während der Fahrt in einer sehr kompakten Fahrtposition, wie in Abbildung 7.3a gezeigt wird. Aus dieser Position heraus wird der Arm durch gezielte Drehungen der einzelnen Gelenke in eine Startposition gebracht. Dieses Vorgehen verhindert Kollisionen mit anderen Aufbauten des Huskys und bietet eine bekannte Ausgangsposition für die Position des Greifers und der Kamera für weitere Bewegungen.
Die Startpositionen unterscheiden sich zwischen dem Anwendungsfall im Flur und in der Aufzugskabine. Im Flur wird davon ausgegangen, dass der Husky erst in Reichweite des Knopfes fahren muss und seine Position gegenüber dem Knopf daher stark variiert. Dies wird durch eine komplexere Anpassung des Greifers an den Winkel zur Wand kompensiert. Im Aufzug hat der Roboterarm nicht viel Platz für ausladende Bewegungen. Dafür ist die Positionierung des Roboters selbst klarer definiert, sodass die Anpassung an die Knopfposition minimiert wird.
Für das Drücken der Knöpfe im Flur wird der Greifer nach vorne rotiert und leicht angehoben, sodass er in einer waagerechten Ausgangsposition ist, wie in Abbildung 7.3b zu sehen ist. Durch horizontale und vertikale Drehung des Greifers wird die Kamera auf den Knopf gerichtet.
In der Aufzugskabine ist die Bewegung des Arms durch die Länge des Greifers eingeschränkt. Mit einer Länge von 32,5 cm verursacht kann eine Rotation des Greifers Kollisionen mit den Aufzugwänden oder nahe stehenden Personen verursachen. Da die Kamera aber nur in Verbindung mit dem Greifer bewegt werden kann, führt eine Suche über alle Seiten der Kabine zwangsläufig zu Bewegungen, die über die Grundfläche des Roboters hinausragen. Es wird das Vorwissen eingebracht, dass der Roboter seitlich zum

Bedienelement steht. Der Arm wird dafür an der Schulter um 90° gedreht, sodass der Greifer nach rechts zeigt, wie es in Abbildung 7.3c zu sehen ist. Anschließend wird der Arm an seiner Schulter in Richtung des Knopfes rotiert und der Greifer waagerecht auf die Höhe des Knopfes gebracht. Die Rotation um die Schulter hat den Vorteil, dass dem Arm von hier aus durch die kombinierten Bewegungen aus Schulter und Ellbogen heraus eine direkte Vorwärtsbewegung auf den Knopf zu gelingt. Mit dieser Maßnahme werden ungewollte Bewegungen, die bei der Planung des kartesischen Pfades entstehen können, minimiert.



(a) Position während der Fahrt

(b) Startposition im Flur

(c) Startposition in der Aufzugkabine

Abbildung 7.3: Darstellung der verschiedenen festgelegten Armposition. In a) ist die kompakte Position während der Fahrt, wie in Kapitel 5 beschrieben. In b) ist die Startposition zum Drücken eines Knopfes im Flur mit nach vorn gerichtetem waagerechtem Greifer. In c) ist die Startposition in der Aufzugskabine. Der Greifer ist in Fahrtrichtung nach rechts auf die Knöpfe gerichtet. Arm und Greifer ragen in dieser Position möglichst wenig über die Plattform hinaus.

7.4 Ansteuern eines erkannten Knopfes

Aus der Startposition heraus beginnt der Prozess der gezielten Bewegung auf den Knopf zu. Der Greifer ist auf den Knopf gerichtet und die Kamera hat den gesuchten Knopf im Bild. Der Bewegungsablauf des Arms auf den Knopf zu unterscheidet sich zwischen Flur und Kabine.
In der Aufzugskabine wird der Greifer direkt auf den Knopf zubewegt, bis sich die Spitze etwa 10 cm vom Knopf entfernt befindet. Um den Versatz der Kameralinse zur Greifer-

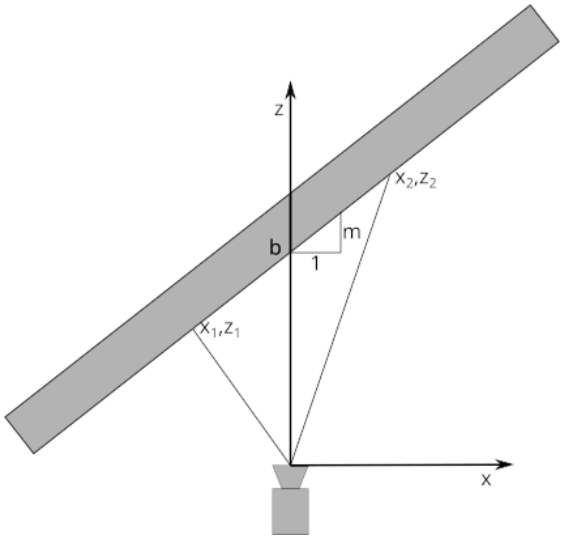


Abbildung 7.4: Draufsicht auf die Kamera mit Blick auf eine schräge Wand. Die Punkte x_1, z_1 und x_2, z_2 werden aus der Punktwolke der Realsense gelesen. Die Front der Wand stellt eine Gerade im Koordinatensystem x,z dar mit dem Achsenabschnitt b und der Steigung m

spitze auszugleichen, wird die Zielposition um 2 cm nach unten und 4 cm nach rechts korrigiert. Für eine exakte Positionsbestimmung wird der Knopf erneut detektiert. Die Geschwindigkeit des Arms wird reduziert und die Spitze des Greifers direkt auf den Knopf zubewegt.

Im Flur wird davon ausgegangen, dass die relative Position des Huskys gegenüber dem Knopf stark variieren kann. Es wird der Winkel zwischen dem Greifer und der Wand ermittelt, sodass der Greifer senkrecht zur Wand ausgerichtet werden kann. Abbildung 7.4 zeigt die Kamera mit Blick auf eine Wand. Gegenüber dem Koordinatensystem der Kamera ist die Wand verdreht. Es werden zwei Punkte x_1, z_1 und x_2, z_2 aus einer horizontalen Ebene der Punktwolke entnommen. Die Front der Wand kann als Gerade gesehen werden, dessen Steigung m berechnet werden kann. Die Gleichungen 7.3 und 7.4 zeigen die Geradengleichungen mit der Steigung m und dem Achsenabschnitt b durch die Punkte (z_1,x_1) und (z_2,x_2).

$$z_1 = m \cdot x_1 + b \tag{7.3}$$

$$z_2 = m \cdot x_2 + b \tag{7.4}$$

$$m = \frac{z_2 - z_1}{x_2 - x_1} \tag{7.5}$$

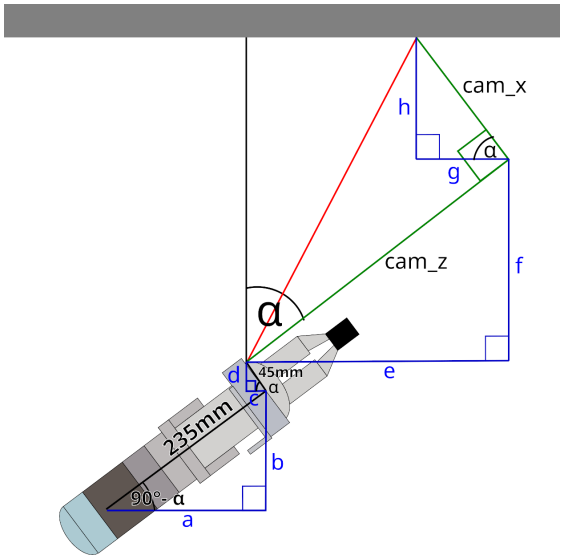


Abbildung 7.5: Darstellung der Entfernungen und Winkel zwischen der Kamera und der Wand im Sichtfeld. Die direkte Distanz (rot) zwischen Kamera und Wand wird als Entfernung in X und Z Richtung gegeben (grün). Aus der Geometrie des Greifers, den Entfernungen cam_z und cam_x und dem Winkel α können die blau dargestellten Distanzen berechnet werden.

Die Umstellung der Gleichungen nach der Steigung m ergibt Formel 7.5. Der Arcustangens der Steigung ist der gesuchte Winkel in Bogenmaß. Der Winkel in Grad umgerechnet ist α.

Wie die Abbildung 7.5 zeigt, kann mit dem berechneten Winkel α festgestellt werden, wie weit der Greifer nach einer Rotation bewegt werden muss, damit seine Spitze an derselben Position verweilt. In Rot ist die Sichtlinie zum Knopf eingezeichnet und in Schwarz die Normale zischen Wand und Kamera. Für die inverse Kinematik muss beachtet werden, dass die Entfernungen zum Knopf, als cam_z und cam_x in Grün dargestellt, von der Kameralinse aus gehen. Da die Rotation um das Handgelenk davon versetzt ist, werden die Abstände in einzelne Komponenten zerlegt, die Anhand bekannter Entfernungen und dem bestimmten Winkel α berechnet werden. Die einzelnen Längen werden in Blau dargestellt. Die Längen a und b ergeben sich aus dem Abstand zwischen Drehpunkt und Mittelpunkt der Kamera. Der Versatz zwischen Kameramittelpunkt und Kameralinse von 45 mm wird in die Längen c und d zerlegt. Zusammen mit den Längen e bis h können die Entfernungen berechnet werden, wie weit der Drehpunkt von dem erkannten Objekt entfernt ist. Anschließend wird der Pfad berechnet und ausgeführt, sodass die Greiferspitze 10 cm senkrecht vom Objekt entfernt zum Stehen kommt. Dabei ist er

leicht zum Mittelpunkt des Objektes versetzt, damit die Kamera den gesamten Knopf im Sichtfeld hat.
Über das gleiche Verfahren kann der Greifer analog auch bei vertikaler Schräglage so an die Wand angepasst werden, dass er senkrecht zur Wand auf Höhe des Knopfes steht. Dies findet Anwendung, wenn der zu drückenden Knopf höher oder niedriger ist als die Höhe in Startposition von etwa 70 cm.
Wie beim Vorgehen in der Aufzugskabine, kann der Greifer nun direkt auf den Knopf zu bewegt werden. Die Greiferspitze trifft am Ende der Bewegung mittig auf den Knopf.

7.5 Erkennung erfolgreicher Tastendrücke

Die Knöpfe geben dem Benutzer beim Betätigen Feedback, indem sie eine geringe Distanz nachgeben und anschließend die Hintergrundbeleuchtung aktiviert wird. Die Erkennung der minimalen Distanz, die der Knopf nachgibt, mit den Tiefensensoren der Realsense zu erfassen, ist nicht möglich. Eine Bewegung rein nach gemessener Entfernung ist sowohl in der Messung mit der Realsense als auch in der Ausführung zu ungenau.
Der Software des Arms bietet ein Notstopp, der bei zu großem Kraftaufwand auslöst. Die Kraft, die der Arm dabei auf ein Objekt ausübt, wird mit einem Versuch gemessen. Es wird eine Kraftmessdose verwendet, die zwischen Greiferspitze und einem Tisch platziert wird. Die Kraftmessdose ist auf eine maximale Zug- oder Druckkraft von 5,5 kg ausgelegt. Abbildung 7.6 zeigt den Kraftverlauf beim gezielten Auslösen des Notstopps bei einer Bewegung, die durch die Rotation um den Ellbogen erzeugt wird. Der Notstopp wird insgesamt drei Mal ausgelöst. Die maximale Kraft, die mit der verwendeten Kraftmessdose gemessen werden kann, beträgt 80 N. Diese Druckkraft wird bei jeder Auslösung des Schutzmechanismus überschritten. Die maximal ausgeübte Kraft des Arms kann mit dem Versuchsaufbau daher nicht ermittelt werden, aber eine Belastung von 80 N ist bereits zu hoch für Interaktionen mit Knöpfen. Es muss eine Lösung gefunden werden, die bei geringerem Kraftaufwand die Bewegung des Arms stoppt.
Die Beleuchtung des Knopfes kann durch die Kamera auf dem Greifer erfasst werden. Während des Kontakts mit dem Knopf ist ein Großteil des Knopfes noch im Bild zu sehen, sodass auf das Aufleuchten reagiert werden kann. Zusätzlich bietet der Arm ein Kraftfeedback über die Kräfte und Momente, die am Endeffektor wirken. Diese können verwendet werden, um eine eigene Kraftobergrenze festzulegen. Im Folgenden werden zwei unabhängige Methoden entworfen, die anhand der beiden Feedback-Möglichkeiten die Bewegung des Arms stoppen, bevor zu viel Kraft ausgeübt wird. Dazu wird die Art

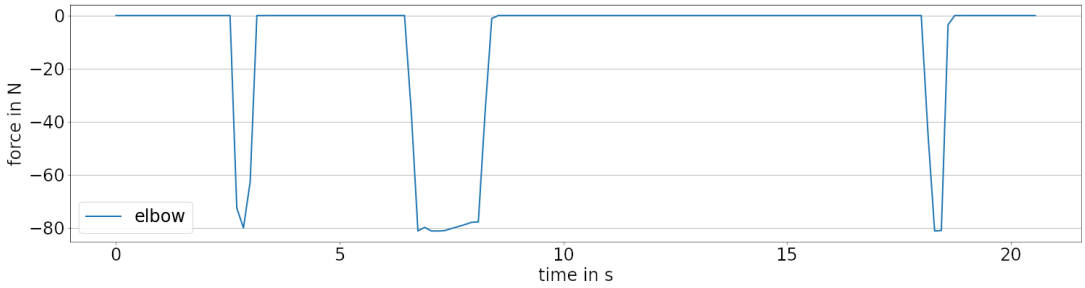


Abbildung 7.6: Kraftverlauf beim wiederholten Auslösen des Notstopps, während der Greifer den Sensor hält. Die Bewegung erfolgt ausschließlich durch Drehung des Ellenbogens. Der Notstopp wird drei Mal ausgelöst und zwischen den Vorgängen zurückgesetzt. Die drei Kraftspitzen im Diagramm zeigen die Drückvorgänge über die Zeit. Die maximal messbare Druckkraft von 80 N wird jedes Mal überschritten.

des Kraft-Feedbacks genauer analysiert und ein Verfahren entwickelt, auf die veröffentlichten Werte zu reagieren.

7.5.1 Kraft Feedback

Der Roboterarm liefert selbst Feedback über die wirkenden Kräfte und Momente am Endeffektor über das ROS-Topic *wrench*. Mit einer Frequenz von 124 Hz werden die Kräfte in X-, Y- und Z-Richtung als auch die Drehmomente um die jeweilige Achse veröffentlicht. Die Abbildung 7.7 zeigt die Position und Orientierung des Koordinatensystems am Endeffektor, an dem der Greifer montiert ist. Um den Achsen eine korrespondierende Krafteinwirkung zuordnen zu können, werden Kraftverläufe aufgezeichnet, während wiederholt in jeweils eine Richtung von Hand Kraft ausgeübt wird. Hierbei geht es nicht um die Kalibrierung der Kräfte, sondern um das allgemeine Verhalten der Kraftwiedergabe sowie die Zuordnung der Achsen zur Wirkungsrichtung.
Der Arm befindet sich in der Startposition mit nach vorne gerichtetem Greifer, wie in 7.3b bereits gezeigt wurde. In einem Versuchsdurchlauf wird drei Mal in dieselbe Richtung auf die Spitze des Greifers gedrückt. Zwischen den Einwirkungen in einem Versuch wird dem Kraftverlauf genug Zeit gelassen, in eine konstante Ruheposition zurückzukehren. Der Durchlauf wird für jede Achse jeweils in Druck- und Zugrichtung wiederholt.
Die Versuche zeigen, dass die Z-Achse die Kräfte in vorwärts und rückwärts gerichteten Kräften wiedergibt. Die Y-Achse zeigt die wirkenden Kräfte bei einer hoch- und runter Bewegung und die X-Achse die Kräfte bei einer links oder rechts gerichteten Bewegung.

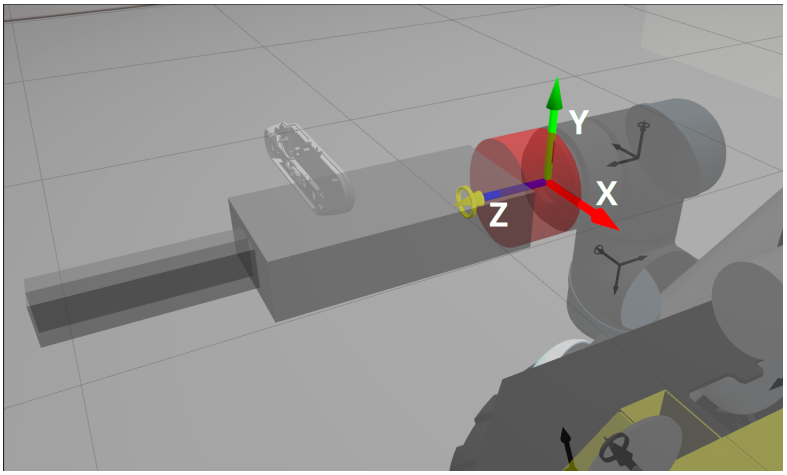


Abbildung 7.7: Endeffektor mit montiertem Greifer in der Simulation. Das Koordinatensystem mit X (rot), Y (grün) und Z (blau) Achse, wie sie im Kraftfeedback verwendet werden.

Wie in Abbildung 7.8 zu sehen ist, verursacht der Druck auf die Greiferspitze einen negativen Ausschlag in der Kraft entlang der Z-Achse. Die beiden anderen Achsen erfahren ebenfalls eine Veränderung, diese fällt jedoch geringer aus.
Über einen Versuchsablauf ist zu beobachten, dass die Ruhekräfte sich zwischen Start und Ende des Versuches verändern. In der Tabelle 7.1 sind die Ruhekräfte vor und nach einem jeweiligen Versuch zusammengefasst. Es sind Unterschiede von bis zu 15 N von Versuchsbeginn bis zum Versuchsende zu verzeichnen. Während die Kräfte entlang der Y-Achse die geringste Schwankung aufweisen, zeigen die beiden anderen Achsen größere Schwankungen, wenn die ausgeübte Kraft entlang der jeweiligen Achse verläuft. Ein ähnliches Verhalten kann bei einer Bewegung des Greifers entlang einer der anderen beiden Achsen beobachtet werden. Die in Tabelle 7.1 aufgelisteten Versuche wurden zeitlich von oben nach unten stehend ausgeführt. Die Ruhekräfte der jeweiligen Achsen vor einem Versuch liegen in etwa bei den Ruhekräften im folgenden Versuch zu Beginn.
Ursache der Änderung in der Ruhekraft ist Art, wie die Kräfte bestimmt werden. In [7] wird beschrieben, wie die Kräfte nicht gemessen werden, sondern anhand der anliegenden Spannung an den Motoren der Gelenke berechnet wird. Diese Berechnung ist ungenauer als eine direkte Messung. Zusätzlich werden die einwirkenden Kräfte am Endeffektor bestimmt und nicht am Greifer, an dem die äußerlichen Kräfte wirken. Das Eigengewicht des Greifers erzeugt eine Last, die in den Messwerten bereits zu sehen ist. Eine Positionsänderung des Greifers in der Halterung zum Endeffektor verursacht die beobachtete

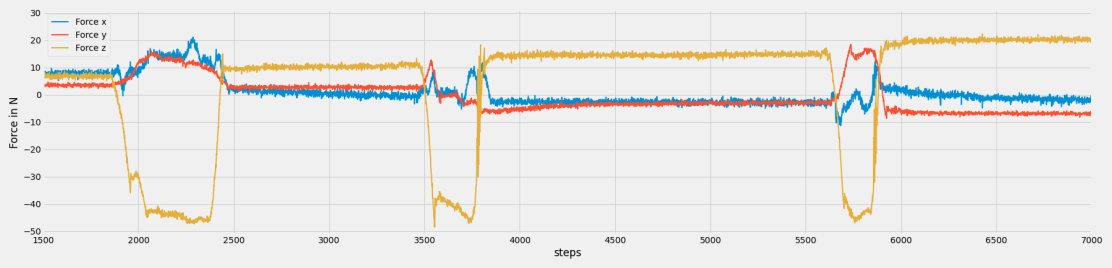


Abbildung 7.8: Kraftmessung der drei Richtungen über das Topic *wrench* bei dreimaligem horizontalen Drücken auf die Spitze des Greifer. Die Z-Achse (gelb) entlang der Wirkungsrichtung zeigt die Krafteinwirkung durch einen negativen Ausschlag. Die Kräfte entlang der Z-Achse ohne äußere Einwirkung steigt um bis zu 12 N zwischen Start und Ende an.

Veränderung in den Ruhekräften.

7.5.2 Kraftstopp

Ziel ist es, anhand des Kraftfeedbacks einen Schwellenwert zu definieren, ab dem die Bewegung abgebrochen werden soll. Betrachtet wird die Kraft in negativer Z-Achse, was einem frontalen Druck auf den Greifer entspricht. Aufgrund der Feststellung in 7.5.1, dass die Kraft bei einem unbelasteten Greifer weder bei 0 N liegt noch zu einem konstanten Wert zurückkehrt, muss die Ruhekraft vor jedem Drücken dynamisch ermittelt werden. Die finale Bewegung auf den Knopf zu wird langsamer ausgeführt als vorherige Bewegungen. Dies lässt genug Zeit, die durchschnittliche Kraft bei Beginn der Bewegung zu bestimmen, bevor der Greifer den Knopf berührt. Die Abbildung 7.9 zeigt die aufgezeichneten Kräfte in Z-Richtung über den Verlauf der verlangsamten Bewegung auf den Knopf zu. Die Kräfte (blau) schwanken während der Bewegung zwischen -20 N und 30 N. Unter der Annahme, dass die Messwerte normalverteilt sind, werden über die ersten 200 Werte Mittelwert und die Standardabweichung bestimmt. Dies entspricht bei 124 Hz einem Zeitraum von 1,61 s. Es wird ein Intervall festgelegt, das der dreifachen Standardabweichung um den Mittelwert entspricht. 99,73 % der auftretenden Messwerte werden in diesem Bereich erwartet. Treten Messwerte außerhalb des Bereichs auf, kann davon ausgegangen werden, dass es sich nicht um die normalen Schwankungen, sondern um eine Einwirkung von außen handelt. Da eine Kraft beim Drücken als negative Kraft auftritt, wird vom Mittelwert das dreifache der Standardabweichung abgezogen. Dieser

Tabelle 7.1: Gemittelte Ruhekräfte vor und nach einem jeweiligen Versuch. Die Versuche wurden von oben nach unten stehend ausgeführt. Die Versuche werden in Druck oder Zug entlang einer Achse unterteilt. Die Kräfte nach einem Versuch bleiben bis zum Beginn des nächsten Versuchs etwa gleich.

Richtung [–]	Zeitpunkt [–]	X [N]	Y [N]	Z [N]
Druck	vor	7.957	3.534	6.790
Z-Achse	nach	-1.925	-6.880	20.050
Zug	vor	-3.485	-6.900	17.549
Z-Achse	nach	-9.500	-8.193	1.838
Rechts	vor	-10.178	-5.677	2.633
X-Achse	nach	-16.690	-5.468	5.769
Links	vor	-19.594	-5.509	6.646
X-Achse	nach	2.221	0.287	2.566
Hoch	vor	3.788	0.915	2.038
Y-Achse	nach	-10.102	0.429	13.965
Runter	vor	-9.676	1.006	12.563
Y-Achse	nach	4.740	-4.830	-0.083

Schwellwert ist in der Abbildung 7.9 als schwarze Linie eingezeichnet.
Damit einzelne Kraftspitzen nicht den Stopp auslösen, wird über die letzten drei gemessenen Werte gemittelt. Liegt deren Mittelwert unterhalb des Schwellwerts, wird die aktuelle Bewegung des Arms unterbrochen. Die grüne senkrechte Markierung in Abbildung 7.9 markiert den Zeitpunkt, ab dem der Stopp ausgelöst wird.
Die maximale Druckkraft, die nach dem Auslösen des Stopps noch erreicht wird, ist in der Abbildung mit einer roten Linie gekennzeichnet. In dem vorliegenden Versuch wurde eine Zeit von etwa 40 ms benötigt, bis die Bewegung des Arms durch das Signal gestoppt wird.
In dieser Zeitspanne kann der Arm eine Druckkraft von fast 90 N aufbringen. Um den Anstieg der Kraft in diesem Zeitraum zu reduzieren, könnte die Bewegungsgeschwindigkeit auf den letzten Zentimetern weiter gesenkt werden. Die Implementation zeigt, dass die Schwankungen mit sinkender Bewegungsgeschwindigkeit nicht zurückgehen.
Aufgrund des ungenauen Feedbacks der Kräfte am Endeffektor treten beim Drücken vereinzelt stärkere Kräfte auf, als erwünscht sind. In Verbindung mit anderen Methoden bietet der Kraftstopp eine gute zweite Linie, die eingreifen kann, falls andere Maßnahmen versagen.

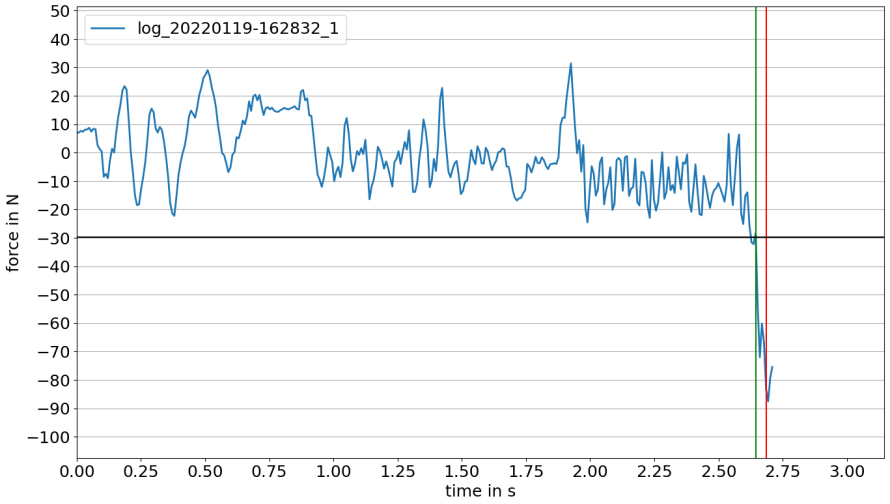


Abbildung 7.9: Beispiel eines Kraftverlaufes (blau) über die langsame Bewegung auf den Knopf zu. Eingezeichnet ist der Schwellwert (schwarz), ab dem der Stopp der Bewegung ausgelöst werden soll. In Grün ist der Zeitpunkt markiert, in dem der Schwellwert überschritten wird. In Rot ist der Zeitpunkt der maximalen Druckkraft nach dem Auslösen des Stopps markiert. Zwischen dem Auslösen des Stopps (grün) und dem tatsächlichen Eintreten (rot) vergehen 40 ms und es wird eine Druckkraft von 87 N erreicht.

7.5.3 Farbstopp

Neben dem haptischen Feedback, bei dem der Knopf eine gewisse Distanz beim Drücken nachgibt, wird dem menschlichen Benutzer durch die aufleuchtende Hintergrundbeleuchtung signalisiert, dass die Interaktion erfolgreich war. Da die Kamera am Greifer beim Drücken einen Großteil des Knopfes noch im Sichtfeld hat, kann das Aufleuchten der Knopfumrandung erfasst werden. Mit Hilfe eines Farbfilters kann das Aufleuchten erkannt und als Signal für den Abbruch der Bewegung verwendet werden.
Es ist davon auszugehen, dass die meisten Aufzüge eine ähnliche Art des Feedbacks verwenden, wobei die Intensität und Farbe je nach Aufzuganlage unterschiedlich sind. Im Falle des Aufzugs in BT7 handelt es sich bei der Beleuchtung um ein orangefarbenes Licht.
Es wird ein Filter erstellt, der diese Beleuchtung im Bild hervorhebt. Durch die Anzahl der nicht gefilterten Pixel kann festgestellt werden, ob die Beleuchtung an ist. Die einzelnen Farbwerte der Beleuchtung des von der Kamera erhaltenen RGB-Bildes werden anhand von Beispielbildern mit aktiver Beleuchtung untersucht. Ein solcher Vergleich ist in Abbildung 7.10 gezeigt. Oben links in der Abbildung ist der Knopf ohne Hinter-

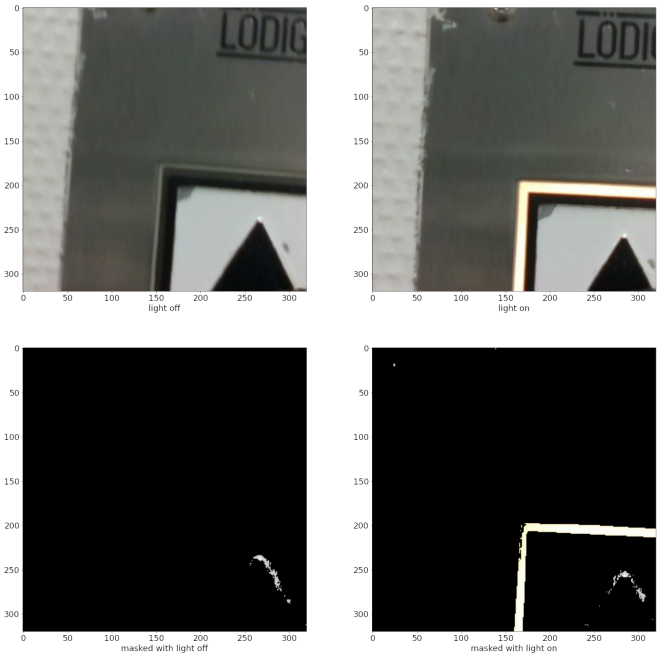


Abbildung 7.10: Vergleich zweier Aufnahmen kurz vor und nach dem Drücken. Vor dem Drücken ist die Beleuchtung aus, nach dem Drücken ist die Beleuchtung an. Darunter sind die beiden Bilder mit der Maske zwischen (120, 140, 200) und (255, 255, 255) für die (R, G, B) Werte. Es werden die Pixel maskiert, die nicht in dem genannten Bereich liegen und damit nicht zu der Beleuchtung gehören. Reflexionen werden stellenweise an der Kante des schwarzen Symbols auf dem Knopf nicht maskiert.

grundbeleuchtung zu sehen und oben rechts mit aktivierter Beleuchtung. Der Vergleich der Histogramme der beiden Bilder in Abbildung 7.11 zwischen aktiver (oben) und nicht aktiver (unten) Beleuchtung zeigt, dass das orangefarbene Licht einen erkennbaren Unterschied in den hellen Pixeln über einem Wert von 200 auslösen. Die Summe der Pixel in diesem Bereich kann als Schwellwert verwendet werden. Der Vergleich mit weiteren Aufnahmen zeigt, dass dieser Schwellwert allein nicht ausreichend ist, um eine eindeutige Entscheidung zu fällen. Durch Reflexionen auf den Armaturen oder bei überbelichteten Bildern kann die Anzahl der hellen Bildpunkte im betrachteten Bereich bereits erhöht sein.
Die Untersuchung der einzelnen farbigen Pixel zeigt, dass die Rotwerte zwischen 120 und 255, die Grünwerte zwischen 140 und 255 liegen und die Blauwerte zwischen 200 und 255. Damit ist der Bereich größer, als bei der Betrachtung des Histogramms festgestellt wurde. Die Erweiterung der betrachteten Werte erzielt im Flur zuverlässige Ergebnisse.

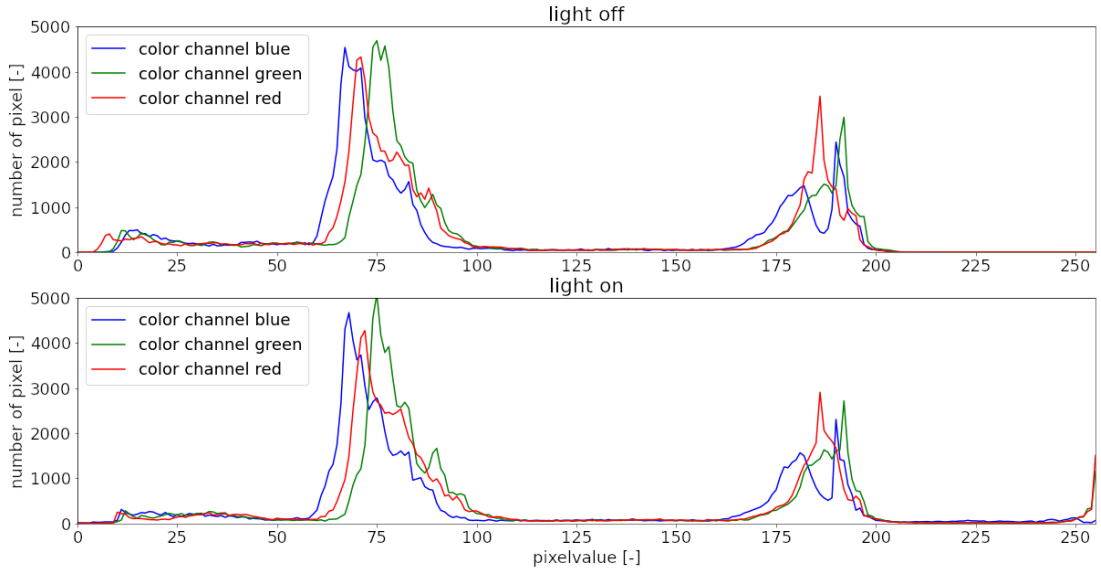


Abbildung 7.11: Histogramm über die drei Farbkanäle bei einer Bildaufnahme mit nicht aktiver Beleuchtung (oben) und aktiver Beleuchtung (unten). Bei der aktiven Beleuchtung ist eine erhöhte Anzahl der Bildpunkte mit Werten über 200 zu sehen. Die Werte der roten und grünen Pixel nimmt oberhalb von 255 stark zu. Die Spitzen um die Werte 75 und 185 herum weisen nur minimale Veränderungen auf.

Die Anwendung des Filters ist in Abbildung 7.10 gezeigt. Unten links ist das maskierte Bild bei nicht aktiver Beleuchtung zu sehen. Bei aktiver Beleuchtung im linken Bild ist die Umrandung des Knopfes deutlich zu erkennen.
Bei veränderten Lichtverhältnissen, wie sie in der Aufzugskabine anzutreffen sind, versagt der Filter. Die Bilder der Kamera sind insgesamt zu dunkel, sodass die Summe der nicht maskierten Bildpunkte unter dem gesetzten Schwellwert bleibt. Der Stopp der Bewegung wird nicht ausgelöst.
Eine alternative Herangehensweise ist, das Bild nicht im RGB-Farbraum, sonder im HSV-Farbraum zu betrachten. Die Trennung in Farbwert (*hue*), Farbsättigung (*saturation*) und Hellwert (*value*) ermöglicht eine bessere Trennung zwischen der farbigen Beleuchtung und den sonst weiß-grauen Armaturen. Der Vergleich mehrerer Aufnahmen, die den gleichen Bildausschnitt mit aktiver und nicht aktiver Beleuchtung zeigen, ergibt einen Schwellwert von 3500 Pixeln. Ist die Anzahl der nicht maskierten Bildpunkte größer als der Schwellwert, wird die Beleuchtung als aktiv gewertet. Mit diesem Verfahren wird erfolgreich ein Abbruch der Armbewegung erreicht, sobald die Beleuchtung des Knopfes aktiviert wird, noch bevor der Kraftstopp eingreifen muss.

Das Verfahren arbeitet unabhängig von der Objekterkennung und kann durch die Wahl eines gezielten Bildausschnittes des 640x480 Pixel großen Bildes mit einer hohen Geschwindigkeit berechnet werden. In der Implementation wurde eine Bildwiederholungsrate von 30 Bilden pro Sekunde gewählt. Für den Einsatz des Farbstopps könnte sich eine Erhöhung der Rate auf 60 Bildern pro Sekunde lohnen, um eine kürzere Reaktionszeit zu erreichen.

8 Fazit

Im Rahmen der autonomen Mobilität im Quartier wurde die autonome Verwendung von Aufzügen durch einen Roboter untersucht. Betrachtet wird dabei der Husky im Gebäude BT7 der HAW. Startpunkt des Szenarios ist der Flur im 7. Stock. Ziel ist die Erkennung und Verwendung der Aufzüge anhand ihrer Bedienelemente.
Der Ablaufplan des gesamten Szenarios wird in fünf Phasen unterteilt. Die einzelnen Phasen umfassen jeweils eine spezifische Aufgabe und sind damit in sich geschlossen, sodass sie bei Fehlern gezielt wiederholt werden können. In der ersten Phase erfasst der Roboter die notwendigen Informationen im Raum über vorhandene Aufzüge und ihre Bedienelemente. In der folgenden Phase kann ein Aufzug gerufen werden, indem der Roboter den entsprechenden Knopf mit dem Greifer an seinem Arm betätigt. Die Position des Knopfes wird durch Objekterkennung mit Realsense Kamera durchgeführt, die sich am Arm des Roboters befindet. Wird in der dritten Phase erkannt, dass der erwartete Aufzug eintrifft, kann dieser betreten werden. Die notwendige Interaktion mit weiteren Fahrgästen in dieser Phase zeigt, dass dem Husky noch die Mittel wie Lampen oder Displays dazu fehlen. Für erste Implementierungen wird daher davon ausgegangen, dass der Roboter alleine anwesend ist. In der vierten Phase befindet sich der Husky im Aufzug und betrachtet die Stockwerksanzeige, um erkennen zu können, wann sein Zielstockwerk erreicht ist. Das Verlassen des Aufzugs bildet die fünfte Phase und den Abschluss des betrachteten Szenarios.
Eine mögliche autonome Navigation mit dem Lidar unter Verwendung von SLAM wird in der Simulation betrachtet. Die erstellte Karte eines 3D-Modells des Flurs zeigt, dass dem Husky in der Aufzugskabine nicht viel Platz zum Manövrieren bleibt, besonders bei weiteren anwesenden Personen. Dies muss beim Einsatz des Roboterarms und der Navigation in und aus dem Aufzug berücksichtigt werden. Für die Implementation des Tastendrückens wird mit dem Roboter vor Ort gearbeitet. Für die Objekterkennung wird das SSD-Mobilenet v2 aus [13] verwendet, das mit selbsterstellten Trainingsdaten in der Lage ist, die Knöpfe im Flur und im Aufzug zu erkennen. Die Ergebnisse zeigen, dass eine Erweiterung der Objekterkennung um die Angaben der Stockwerke auf Displays

oder Schildern im Flur vielversprechend ist. Dank des sechsgelenkigen Arms kann der Greifer beim Ansteuern eines Knopfes so ausgerichtet werden, dass er unabhängig von der Position des Roboters senkrecht auf den Knopf drückt. Dafür werden die Daten aus der Punktwolke verarbeitet, die von der Realsense Kamera geliefert werden.
Um Beschädigungen an den Bedienelementen zu vermeiden, die durch übermäßig starkes Drücken des Arms verursacht werden können, werden zwei Mechanismen zum Stoppen des Arms entwickelt. Als erste Methode wird das Feedback des Arms verwendet, das die Kräfte am Endeffektor anhand der anliegenden Spannungen an den Motoren der Gelenke berechnet. Mit diesen Werten kann der Arm rechtzeitig gestoppt werden, bevor zu viel Kraft aufgewendet wird. Durch die Ungenauigkeiten im Feedback kommt es bei dieser Methode noch teilweise zu gröberem Kontakt als erwünscht ist. In der zweiten Methode wird die aufleuchtende Hintergrundbeleuchtung eines Knopfes mit der Kamera erfasst und die Bewegung des Arms gestoppt.
Für den zukünftigen Einsatz könnte es vorteilhaft sein, einen Adapter zu entwickeln, der den Kontakt zwischen Greifer und Objekt abfedert. Da der Greifer bereits vorhanden ist, kann der Adapter so gestaltet werden, dass er für die benötigten Aufgaben gegriffen wird und anschließend wieder auf dem Roboter verstaut werden kann.

Literaturverzeichnis

[1] Nguyen, Hai ; Deyle, Travis ; Reynolds, Matt S. ; Kemp, Charles C.: PPS-tags: Physical, Perceptual and Semantic tags for autonomous mobile manipulation Georgia Institute of Technology, 2009

[2] Sani, Mohammad F. ; Karimian, Ghader: Automatic navigation and landing of an indoor AR. drone quadrotor using ArUco marker and inertial sensors. In: *2017 International Conference on Computer and Drone Applications (IConDA)*, 2017, S. 102–107

[3] Wang, Fan ; Chen, Gerry ; Hauser, Kris: Robot Button Pressing in Human Environments. In: *2018 IEEE International Conference on Robotics and Automation (ICRA)*, Institute of Electrical and Electronics Engineers Inc., 2018 (Proceedings - IEEE International Conference on Robotics and Automation), S. 7173–7180

[4] Chiu, Yu-Chen ; Tsai, Chi-Yi ; Ruan, Mind-Da ; Shen, Guan-Yu ; Lee, Tsu-Tian: Mobilenet-SSDv2: An Improved Object Detection Model for Embedded Systems. In: *2020 International Conference on System Science and Engineering (ICSSE)*, 2020, S. 1–5

[5] Perigis, Stephan ; Tiedemann, Tim ; Maximilian A., De M.: *Test Area Intelligent Quartier Mobility (TIQ)*. https://autosys.informatik.haw-hamburg.de/project/smartmobility/, Abruf: 07.02.2022

[6] Inc., Clearpath R.: *Husky Unmanned Ground Vehicle Robot*. https://clearpathrobotics.com/husky-unmanned-ground-vehicle-robot/, Abruf: 07.02.2022

[7] Kufieta, Katharina: *Force Estimation in Robotic Manipulators: Modeling, Simulation and Experiments*. 2014

[8] Padovano, Antonio ; Longo, Francesco ; Nicoletti, Letizia ; Mirabelli, Giovanni: A Digital Twin based Service Oriented Application for a 4.0 Knowledge Navigation in the Smart Factory. In: *IFAC-PapersOnLine* 51 (2018), Nr. 11, 631-636. https://www.sciencedirect.com/science/article/pii/S2405896318315143. – ISSN 2405–8963. – 16th IFAC Symposium on Information Control Problems in Manufacturing INCOM 2018

[9] Robotics, Clearpath: *Husky Gmapping Demo*. https://www.clearpathrobotics.com/assets/guides/kinetic/husky/HuskyGmapping.html, Abruf: 21.02.2022

[10] Troniak, Daniel ; Sattar, Junaed ; Gupta, Ankur ; Little, James J. ; Chan, Wesley ; Calisgan, Ergun ; Croft, Elizabeth ; Loos, Machiel Van d.: Charlie Rides the Elevator – Integrating Vision, Navigation and Manipulation towards Multi-floor Robot Locomotion. In: *2013 International Conference on Computer and Robot Vision*, 2013, S. 1–8

[11] Li, Shuang ; Chen, Yujing ; Meng, Yuhao ; Lou, Yunjiang: Autonomous Elevator Button Recognition and Operation Framework for Multi-Floor Mobile Manipulator Navigation. In: *2021 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, 2021, S. 383–388

[12] Howard, Andrew G. ; Zhu, Menglong ; Chen, Bo ; Kalenichenko, Dmitry ; Wang, Weijun ; Weyand, Tobias ; Andreetto, Marco ; Adam, Hartwig: *MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications*. 2017

[13] Liu, Wei ; Anguelov, Dragomir ; Erhan, Dumitru ; Szegedy, Christian ; Reed, Scott ; Fu, Cheng-Yang ; Berg, Alexander C.: SSD: Single Shot MultiBox Detector. In: *Lecture Notes in Computer Science* (2016), 21–37. http://dx.doi.org/10.1007/978-3-319-46448-0_2. – ISBN 9783319464480

[14] Lin, Tsung-Yi ; Maire, Michael ; Belongie, Serge ; Bourdev, Lubomir ; Girshick, Ross ; Hays, James ; Perona, Pietro ; Ramanan, Deva ; Zitnick, C. L. ; Dollár, Piotr: *Microsoft COCO: Common Objects in Context*. 2015

[15] Consortium, COCO: *Common Objects in Context - Detection Evaluation*. https://cocodataset.org/#detection-eval, Abruf: 16.02.2022

Glossar

Berliner Tor 7 Eines der Gebäude auf dem Hauptcampus der HAW Hamburg in dem das Dekanat der Fakultät Technik und Informatik sowie die Departments Informatik, Informations- und Elektrotechnik sind.

HAW Hamburg Die HAW Hamburg ist die vormalige Fachhochschule am Berliner Tor.

Erklärung zur selbstständigen Bearbeitung einer Abschlussarbeit

Hiermit versichere ich, dass ich die vorliegende Arbeit ohne fremde Hilfe selbständig verfasst und nur die angegebenen Hilfsmittel benutzt habe. Wörtlich oder dem Sinn nach aus anderen Werken entnommene Stellen sind unter Angabe der Quellen kenntlich gemacht.

Ort	Datum	Unterschrift im Original