

Teleoperation und kontinuierliches Lernen für cyber-physische Systeme am Beispiel eines Kommissionierroboters

Zur Erlangung des akademischen Grades eines

Doktors der Ingenieurwissenschaften (Dr.-Ing.)

von der KIT-Fakultät für Maschinenbau
des Karlsruher Instituts für Technologie (KIT)

angenommene

DISSERTATION

von

M.Sc. Jonathan Auberle
geb. Dziedzitz in Konstanz

Tag der mündlichen Prüfung:

06.06.2025

Hauptreferent:
Korreferentin:

Prof. Dr.-Ing. Kai Furmans
Prof. Dr.-Ing. Barbara Deml

Kurzfassung

Der steigende Einsatz von Automatisierungstechnik für komplexe Logistikaufgaben, wie beispielsweise die automatisierte Kommissionierung von Einzelteilen, erfordert robuste cyber-physische Systeme (CPS). Neue und unbekannte Aufgabenstellungen stellen heutige Systeme vor große Herausforderungen, führen zu Systemausfällen und geringer Akzeptanz bei Mitarbeiter*innen.

In dieser Arbeit wird ein gültiges Konzept für das kontinuierliche Lernen von CPS durch Teleoperation vorgestellt. Durch intuitives Miteinbeziehen menschlicher Fähigkeiten wird eine Bedienung durch Teleoperation ohne notwendige Vorkenntnisse ermöglicht. Die erarbeitete Kombination aus automatisierter Funktion und Teleoperation beugt drohenden Systemausfällen vor und erlaubt einen Einsatz von CPS in neuen Einsatzszenarien. Während der Teleoperation werden Lerndaten generiert, welche die autonome Leistungsfähigkeit des CPS kontinuierlich steigern. Die entkoppelte Hilfestellung des Menschen wird in vier eigens entwickelte Teleoperationsstufen eingeteilt. Diese beziehen sich auf die eingesetzte Perzeption und die Art der Verbindung zwischen Teleoperator und CPS. Das entwickelte Konzept wird beispielhaft für den Einsatz mit einem Kommissionierroboter umgesetzt. Anhand eines eigenständig entwickelten Demonstrators werden notwendige Module ausführlich vorgestellt und diskutiert. Eine Probandenstudie mit 42 Teilnehmer*innen gibt Aufschluss über den Zusammenhang von Latenz und Aufgabenleistung. Weitere wichtige Einflussfaktoren für eine Teleoperation in der virtuellen Realität (VR) werden identifiziert und untersucht. Anhand von 1000 automatisierten und teleoperierten Kommissionierungen eines unbekannten Artikelspektrums wird der positive Einfluss des vorgestellten Konzepts demonstriert. Durch den Einsatz der vier entwickelten Stufen der Teleoperation kann eine Erfolgsquote von 100% erreicht werden. Die generierten Lerndaten führen zu einer signifikanten Reduzierung von Fehlerkennungen und erhöhen nachhaltig die automatisierte Leistungsfähigkeit des Kommissionierroboters.

Abstract

The increasing use of automation technology for complex logistics tasks, such as automated picking of individual parts, requires robust cyber-physical systems (CPS). New and unknown tasks pose major challenges to today's systems, leading to system failures and low acceptance by employees.

In this dissertation one valid concept for continuous learning of CPS through teleoperation is presented. By intuitively incorporating human capabilities, operation by teleoperation is enabled without necessary prior knowledge. The developed combination of automated function and teleoperation prevents imminent system failures and allows the use of CPS in new application scenarios. During teleoperation, learning data is generated that continuously increases the autonomous performance of the CPS. Decoupled human assistance is classified into four dedicated teleoperation levels. These relate to the perception used and the type of connection between the teleoperator and the CPS. The developed concept is exemplarily implemented for the use with a picking robot. Using an independently developed demonstrator, necessary modules are presented and discussed in detail.

A study with 42 participants provides information about the correlation between latency and task performance. Further, important influencing factors for teleoperation in virtual reality (VR) are identified and investigated.

The positive influence of the presented concept is demonstrated based on 1000 automated and teleoperated picks of an unknown range of items. By using the four developed stages of teleoperation, a success rate of 100% can be achieved. The generated learning data leads to a significant reduction of error detections and sustainably increases the automated performance of the picking robot.

Inhaltsverzeichnis

Kurzfassung	i
Abstract	iii
1 Einleitung	1
1.1 Problembeschreibung und Fragestellungen	1
1.2 Struktur der Arbeit	3
2 Stand der Wissenschaft	5
2.1 Definitionen und Begrifflichkeiten	5
2.2 Kommissioniersysteme	6
2.3 Menschliche Fähigkeiten	11
2.4 Teleoperation und Autonomie	14
2.5 Fazit und Forschungslücke	22
3 Stand der Technik	25
3.1 Mathematische Grundlagen	25
3.1.1 Koordinatensysteme	26
3.1.2 Rotation im Raum	28
3.1.3 Roboterkinematik	33
3.2 Netzwerkkommunikation	35
3.2.1 Netzwerkschichten	35
3.2.2 Definitionen und Begrifflichkeiten	37
3.2.3 Internet Protokoll (IP)	39
3.3 Datenkompression	41
3.3.1 Verlustfreie Kompression	44
3.3.2 Verlustbehaftete Kompression	45
3.3.3 Kompression von Videodaten	48

3.4	Anomalie-Erkennung	49
3.5	Kameramodell	52
3.6	Maschinelles Lernen	55
4	Konzept für Teleoperation und kontinuierliches Lernen	61
4.1	Vorstellung des erarbeiteten Konzepts	61
4.2	Automatisierte Funktion	64
4.3	Anomalieerkennung	64
4.4	Teleoperation	65
4.5	Lerndatengenerierung	65
4.6	Lernen	65
4.7	Verbindung von Mensch und Maschine	66
5	Umsetzung der Teleoperation	71
5.1	Vorstellung des entwickelten Demonstrators	71
5.2	Teleoperationsmodule	73
5.2.1	Graphische Benutzeroberfläche	73
5.2.2	Datenaufnahme	77
5.2.3	Datenkompression	83
5.2.4	Datenübertragung	88
5.2.5	Dekompression und Visualisierung	92
5.2.6	Robotersteuerung	97
6	Umsetzung der Automatisierten Funktion, Anomalieerkennung und Lerndatengenerierung	99
6.1	Objekterkennung	99
6.2	Greifpunktbestimmung	101
6.3	Ablageposition	103
6.4	Anomalieerkennung	104
6.5	Lerndatengenerierung	106
7	Versuchsdurchführung, Funktionsdemonstration und Bewertung	111
7.1	Systemlatenz	112
7.2	Proband*innenstudie über den Einfluss von VR und Latenz	117

7.2.1	Qualitativer Einfluss von Latenz auf die Teleoperationsdauer	125
7.2.2	Einfluss der Visualisierung in VR	128
7.2.3	Lernfähigkeit der Proband*innen	132
7.2.4	Quantitative Bewertung und Modellierung des Zusammenhangs von Teleoperationsdauer und Latenz für Pick-and-Place Aufgaben	135
7.3	Lernen durch Teleoperation	143
7.3.1	Versuchsaufbau	144
7.3.2	Versuchsdurchführung und Versuchsauswertung	147
8	Zusammenfassung und Fazit	155
9	Anhang	157
9.1	Trainingsparameter	157
9.2	Datenkompression	160
10	Studentische Arbeiten	161
Literaturverzeichnis		173
Eigene Veröffentlichungen		195
	Journalartikel	195
	Konferenzbeiträge	195

1 Einleitung

1.1 Problembeschreibung und Fragestellungen

Der Einsatz von Automatisierungstechnik ist in den letzten Jahren weltweit gestiegen, allerdings wird der Großteil der eingesetzten Roboter nach wie vor für repetitive Aufgaben, wie z.B. in der Automobilindustrie, eingesetzt [IFR(2019)], [IFR(2020)].

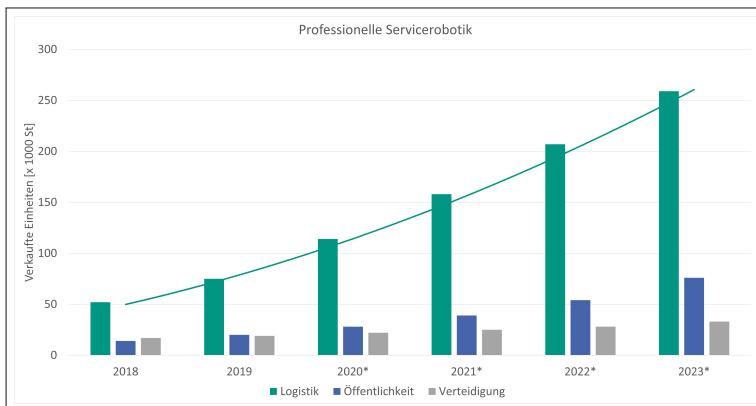


Abbildung 1.1: Ist-Werte und Prognose(*) über die weltweiten Absätze von professionellen Service-Robotern [IFR(2020)]

In den kommenden Jahren ist laut aktuellen Prognosemodellen [IFR(2020)] mit einem starken Anstieg von professionellen Servicerobotern in der Logistik und in der Öffentlichkeit zu rechnen (Abbildung 1.1). Professionelle Serviceroboter sind Dienstleistungsroboter, die außerhalb des privaten Umfelds eingesetzt werden. Aufgaben wie der Transport von Paketen, die Handhabung von Objekten oder eine audiovisuelle Museumsführung sind einige Anwendungsbeispiele.

Häufig werden Serviceroboter mit einer Kombination aus modernster Aktorik, Sensorik und Algorithmen aus dem Bereich des maschinellen Lernens bzw. der künstlichen Intelligenz (KI) eingesetzt. Diese Kombination ermöglicht ein deutlich breiteres Anwendungsspektrum als bisherige, starre Industrierobotik.

Bisher ist ein flächendeckender Einsatz, insbesondere im deutschen Mittelstand, nicht zu beobachten [accenture und Welt(2020)]. Hauptgrund sind neben hohen Anschaffungskosten und Bedenken beim Datenschutz vor allem mangelnde Fachkräfte und eine hohe Störanfälligkeit der Systeme.

Langfristige Digitalisierungsziele der deutschen Bundesregierung werden durch Innovationsprogramme des Bundesministeriums für Bildung und Forschung (BMBF), wie z.B. die *Hightech-Strategie 2025*, vorangetrieben. Das Bundesministerium für Wirtschaft und Energie (BMWi) stellt durch weitere Innovationsprogramme, wie *Smarte Datenwirtschaft*, *Smart Service Welt* und *PAiCE - Digitale Technologien für die Wirtschaft*, einen zügigen Transfer zwischen Forschung und Industrie sicher.



Abbildung 1.2: Forschungsprojekt QBIIK, im Rahmen des PAiCE Technologieprogramms des Bundesministerium für Wirtschaft und Energie (BMWi). Autonome Logistikplattform mit intuitiver Mensch-Maschine-Schnittstelle, eigene Darstellung

Die vorliegende Dissertation knüpft unmittelbar an die Forschungsergebnisse von QBIIK [BMWi(2020a)] an, das vom BMWi im Rahmen des PAiCE Technologieprogramms gefördert wurde. Ziel des Forschungsvorhaben QBIIK war die Entwicklung eines automatisierten Logistikfahrzeugs mit Industrieroboter (Abbildung 1.2), das sich ohne jeglichen Eingriff auf die Infrastruktur (in der bisher lediglich Menschen arbeiten) zurechtfindet.

Über die Inhalte des Forschungsprojekts hinaus soll die vorliegende Dissertation eine entkoppelte Bedienung bzw. Steuerung (Teleoperation) komplexer cyber-physischer Systeme (CPS) am Beispiel eines stationären Kommissiernierroboters, also eines professionellen Serviceroboters, behandeln. Für die Steuerung sollen keine Vorkenntnisse aus den Bereichen Informatik, Robotik oder Automatisierungstechnik benötigt und somit eine Lösung für mangelnde Fachkräfte geboten werden. Weiterhin soll die erwähnte Störanfälligkeit durch eine Verknüpfung von menschlichem Fachwissen und automatisierter Funktion adressiert und untersucht werden.

Es wird ein Konzept erarbeitet, das durch die Integration des Menschen neue Anwendungsbereiche der Robotik erschließen kann und den Einsatz in bereits bestehende Anwendungen intuitiver und robuster gestaltet.

1.2 Struktur der Arbeit

Die vorliegende Arbeit formuliert, ausgehend von einer *Einleitung* (1) in die Thematik und einer Übersicht über den *Stand der Wissenschaft* (2), insgesamt sechs forschungsbegleitende Fragestellungen, die in Unterkapitel 2.5 vorgestellt werden.

Anschließend werden die notwendigen Grundlagen aus den Bereichen Robotik, Netzwerktechnik und künstliche Neuronale Netze im *Stand der Technik* (3) anschaulich beschrieben.

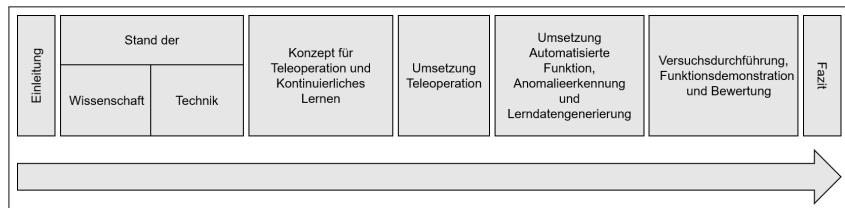


Abbildung 1.3: Aufbau der vorliegenden Dissertation, eigene Darstellung

Das Kapitel *Konzept für Teleoperation und Kontinuierliches Lernen* (4) stellt ein einheitliches Framework für die Verbindung von automatisierter Funktion eines CPS mit den menschlichen Fähigkeiten durch Teleoperation vor.

Fokus des Frameworks ist eine mittelbare und unmittelbare Lernfähigkeit des CPS durch verschiedene Stufen der Teleoperation.

Die beiden folgenden Kapitel *Umsetzung der Teleoperation* (5) und *Umsetzung Automatisierte Funktion, Anomalieerkennung und Lerndatengenerierung* (6) beschreiben eine beispielhafte Implementierung des erarbeiteten Frameworks am Beispiel eines Kommissionierroboters.

Das letzte Kapitel *Versuchsdurchführung, Funktionsdemonstration und Bewertung* (7) untersucht die Leistungsfähigkeit der entwickelten Teleoperationslösung, den Einfluss von Latenz während einer Teleoperation sowie die Lernfähigkeit und die damit einhergehende Leistungssteigerung eines CPS durch Teleoperation. Die entwickelte Lösung sowie die erhaltenen Ergebnisse werden im *Fazit* (8) diskutiert und zusammengefasst.

2 Stand der Wissenschaft

2.1 Definitionen und Begrifflichkeiten

Bevor auf Stand der Wissenschaft eingegangen wird, sollen einige grundsätzliche Begrifflichkeiten vorgestellt und definiert werden, die in dieser Arbeit Anwendung finden.

Cyber-physisches System Nach der Acatech Studie AgendaCPS [Geisberger und Broy(2012)] verbinden cyber-physische Systeme die physikalische und informationstechnische Welt durch ein Zusammenspiel von eingebetteten Systemen, Anwendungssystemen und Infrastrukturen. Hierunter fällt insbesondere die internetbasierte Vernetzung und Integration der Mensch-Technik Interaktion.

Teleoperation und Teleoperator Teleoperation bezeichnet die Steuerung eines (cyber-physischen) Systems aus der Ferne. Der steuernde Mensch wird hierbei als *Teleoperator* bezeichnet.

Virtual Reality (VR) Virtual Reality oder kurz VR beschreibt eine immersive Interaktion des Menschen mit einer computergenerierten Simulation. Hierbei kommen verschiedene VR-Geräte, wie z.B. eine VR-Brille (engl. head-mounted-display, HMD) oder ein VR-Controller (Joystick) zum Einsatz.

Roboter Der Begriff Roboter ist im deutschen Sprachgebrauch breit gefächert. Er dient zur Beschreibung von Fahrerlosen Transportfahrzeugen (FTF), Knickarmrobotern, selbstfahrenden Gabelstaplern, automatischen Staubsaugern und sogar Humanoiden. Des Weiteren findet häufig eine Abgrenzung zwischen

Industrieroboter, kollaborierenden Roboter (engl. Abk. Cobots) und Serviceroboter statt. Innerhalb dieser Dissertation werden die Begriffe folgendermaßen voneinander abgegrenzt:

- **Roboter:** Alleinstehend soll der Begriff Roboter als Knickarmroboter verstanden werden.
- **mobiler Roboter:** Beschreibt selbständig navigierende Fahrzeuge unabhängig von ihrer Größe, umfasst sowohl FTF als auch selbstfahrende Gabelstapler.
- **Knickarmroboter:** Umfasst Industrieroboter und kollaborierende Knickarmroboter.
- **Robotiksystem:** Knickarmroboter mit einem Effektor (z.B. Greifer) sowie diverser Sensorik.

2.2 Kommissioniersysteme

Die Kommissionierung beschreibt in der Logistik eine auftragsbasierte Zusammenstellung von einem oder mehreren Artikeln aus einem gegebenen Sortiment. Eingesetzte Kommissioniersysteme lassen sich grundsätzlich in zwei Prinzipien einteilen [Arnold und Furmans(2019)]:

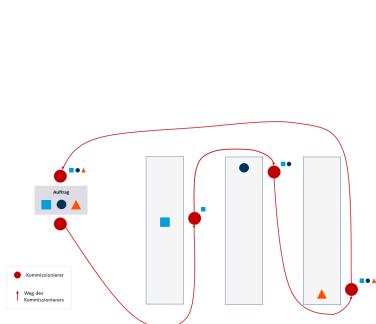
- Person-zu-Ware
- Ware-zu-Person

Person-zu-Ware Die einstufige Person-zu-Ware Kommissionierung beschreibt die vollständige Auftragskommissionierung durch eine Person. Der Auftrag wird an die zu kommissionierende Person übergeben und diese bewegt sich daraufhin im Warenlager. Dort werden lediglich die (von Kunden*innen) bestellten Artikel entnommen und anschließend für den Versand vorbereitet.

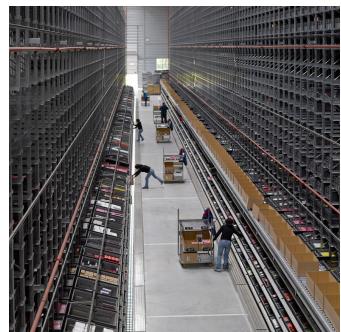
Bei der mehrstufigen Person-zu-Ware Kommissionierung werden nicht nur die Artikel eines einzelnen Auftrages, sondern die mehrerer Aufträge gemeinsam kommissioniert und anschließend sortiert und konsolidiert. Das spart

Laufwege zwischen den Entnahmen und erlaubt einen höheren Durchsatz bei entsprechender Bestellmenge.

Die Person-zu-Ware Kommissionierung kann entweder ohne Hilfsmittel durchgeführt werden (kommissionierende Person ist zu Fuß unterwegs) oder es werden verschiedene Flurförderfahrzeuge, Gabelstapler oder Senkrechtstapler für die Handhabung von Ladungsträgern oder der Bedienung eines Hochregallagers eingesetzt. Das Prinzip ist in Abbildung 2.2a dargestellt und Abbildung 2.1b zeigt eine reale Anwendung mit Hilfe eines Kommissionierwagens.



(a) Prinzip der Person-zu-Ware Kommissionierung
[SelectLine(2020)]



(b) Beispiel einer Person-zu-Ware Kommissionierung mit Hilfe eines Kommissionierwagens der Firma Ulma [UL-MA(2020)]

Abbildung 2.1: Prinzip und Beispiel der Person-zu-Ware Kommissionierung

Ware-zu-Person Im Gegensatz zur Person-zu-Ware Kommissionierung, bei der sich die Person durch das Warenlager bewegt, werden bei der Ware-zu-Person Kommissionierung die auftragsspezifischen Artikel an den Arbeitsplatz des Kommissionierers befördert.

Häufig werden hierfür Stetigförderer in Form von Rollenförderern eingesetzt, die verschiedene Kleinladungsträger (KLTs) aus einem automatisierten (Kleinteile-)Lager an den Arbeitsplatz der kommissionierenden Person und zurück in das Lager befördern. Diese entnimmt dort möglichst ergonomisch die bestellten Artikel und die übrigen werden durch den Stetigförderer zurück in das Lager transportiert. Das Prinzip ist in Abbildung 2.2a dargestellt sowie

eine am Markt erhältliche Variante der Firma Dematic [Klerk und Fran(2017)] in Abbildung 2.2b

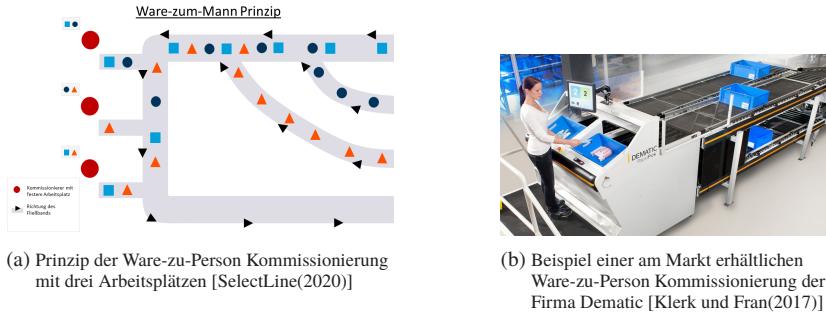
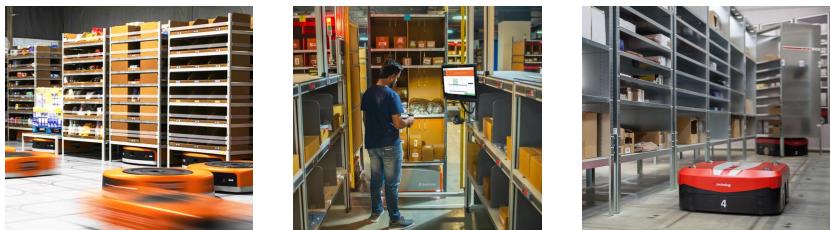


Abbildung 2.2: Prinzip und Beispiel der Ware-zu-Person Kommissionierung

Einen Sonderfall stellen die Systeme Kiva von Amazon Robotics [Wessling(2022)], der RangerGTP von GreyOrange [GreyOrange(2021)] oder das CarryPick System von Swisslog [Swisslog(2022)] dar (Abbildung 2.3). Beide Systeme transportieren ganze Regale an den Arbeitsplatz der kommissionierenden Person. Hierfür werden fahrerlose Transportsysteme eingesetzt, die das zu transportierende Regal unterfahren, mit einem Hubmechanismus anheben und das Regal an die entsprechende Kommissionierstation transportieren. Nach der Artikelentnahme wird das Regal weiter transportiert. Entweder zurück in das Lager oder direkt an die nächste Kommissionierstation. Die Koordination der Systeme wird von einem zentralen Computersystem gesteuert.

Alle bisher vorgestellten Kommissioniersysteme setzen auf die Fähigkeiten des Menschen bei der Kommissionierung. Durch verbesserte Sensorik, neue Algorithmen und GPU-basierter Bildverarbeitung konnten in den letzten Jahren erhebliche Verbesserungen in der Kommissionierung durch Roboter erzielt werden [3], [Jaghbeer et al.(2020)Jaghbeer, Hanson, und Johansson].



(a) Amazon Robotics (ehemalig
Kiva) [Wessling(2022)]

(b) GreyOrange RangerGTP
[GreyOrange(2021)]

(c) Swisslog CarryPick [Co-
merge(2022)]

Abbildung 2.3: Sonderfall der Ware-zu-Person Kommissionierung. Die mobilen Roboter transportieren ganze Regale zu verschiedenen Kommissionierstationen, bei denen die gewünschten Artikel händisch entnommen werden

Die Komponente *Roboter* gewinnt im logistischen Umfeld kontinuierlich an Bedeutung und inzwischen ist immer häufiger die Rede von zwei weiteren Kommissionierprinzipien:

- Ware-zu-Roboter
- Roboter-zu-Ware

Ware-zu-Roboter Die Ware-zu-Roboter Kommissionierung ist vergleichbar mit dem Prinzip der *Ware-zu-Person-Kommissionierung*. Die Integration in ein bereits bestehendes System lässt sich (zumindest aus mechanischer Sicht) ohne große Hürden realisieren. Aus technologischer Sicht ist ein Einsatz unter kontrollierten Bedingungen realistisch umzusetzen.

Abbildung 2.4 zeigt den Roboter *IFL PiRo*, der am Karlsruher Institut für Technologie (KIT) für die Amazon Robotics Challenge (ARC) [Colling et al.(2018)Colling, Dziedzitz, Furmans, Hopfgarten, und Markert] entwickelt wurde. Das System verwendet eine Vielzahl unterschiedlicher Verfahren für die Erkennung und Handhabung bekannter Objekte aus statischen Ladungsträgern [Colling et al.(2017)Colling, Dziedzitz, Hopfgarten, und Markert et al.]. Während die Robotersysteme damals fast ausschließlich aus Forschungsprojekten und Machbarkeitsstudien bestanden, sind inzwischen einige kommerzielle Systeme am Markt erhältlich [Technote(2022)], die insbesondere auf die Erkennung und Handhabung von unbekannten Objekten abzielen.



(a) Forschungsprojekt IFL
PiRo bei der Amazon Robotics Challenge [Colling et al.(2017)] [Colling, Dziedzitz, Hopfgarten, und Mankert et al.]



(b) Kommerzielle Bin Picking
Lösung von Viscarious
[Viscarious(2021)]



(c) Kommerzielle Bin Picking
Lösung Covariant AI
[Covariant(2022)]

Abbildung 2.4: Beispiele der Ware-zu-Roboter Kommissionierung

Die Umsetzung von Viscarious [Viscarious(2021)] kommissioniert mit Hilfe von mehreren Robotern diverse Objekte aus einem statischen Ladungsträger auf ein bewegtes Förderband. Das System von Covariant AI [Covariant(2022)] verwendet Stetigfördertechnik für den An- und Abtransport der Ladungsträger und platziert den Roboter mittig.

Roboter-zu-Ware Das Prinzip der Roboter-zu-Ware Kommissionierung ist im Vergleich zur *Ware-zu-Roboter Kommissionierung* deutlich komplexer. Ein Robotiksystem muss mit vergleichbarer Handhabungstechnologie und Sensorik ausgestattet und zusätzlich mobil sein. Viele Projekte im Bereich der Roboter-zu-Ware Kommissionierung sind Prototypen oder Bestandteil aktueller Forschung.

Abbildung 2.5 zeigt die Pilotierung des mobilen Roboters QBIK [5] im Warensupermarkt von Audi Sport [AUDI(2022)] in Neckarsulm. QBIK wurde gemeinsam am KIT mit der Firma Bär Automation [Bär(2022)] und STILL [STILL(2022)] entwickelt. Auch im Bereich der *Roboter-zu-Ware Kommissionierung* finden sich am Markt bereits kommerzielle Anbieter, wie z.B. das System Swift von IAM Robotics [IAM(2021)] oder Soto von Magazino [Magazino(2022b)], die eine Kombination aus FTF und Knickarmroboter anbieten. Meist sind diese jedoch für spezielle Einsatzszenarien ausgelegt.



Abbildung 2.5: Beispiele der Roboter-zu-Ware Kommissionierung

Der Faktor Mensch Trotz etlicher Jahre Forschung und Entwicklung in den Bereichen mobiler Robotik und automatisierter Kommissionierung finden sich in der Realität wenig bis keine Systeme in dynamischen bis hochdynamischen Umgebungen. Der immense technologische Aufwand, der zur Unterstützung des Menschen bei der Kommissionierung aufgewendet wird, ist ein guter Indikator für die anhaltende Überlegenheit des Menschen in dynamischen Logistikprozessen.

Technologien wie Pick-by-Voice, Pick-by-Light, Put-by-Light, Pick-by-Spot, Pick-by-Vision, etc. sind Technologien, die den Menschen, z.T. mit sehr komplexen Algorithmen, unterstützen sollen. Entweder steht die Ergonomie des Menschen, die Fehlertoleranz oder die Effizienz bei der Aufgabenerfüllung im Vordergrund (Abbildung 2.6).

Algorithmen für Prozessverständnis, Objekterkennung, Orientierung im Raum und viele weitere sind hierbei im Einsatz. Dennoch wird die Handhabung weitestgehend vom Menschen übernommen.

2.3 Menschliche Fähigkeiten

Der Mensch besitzt eine Vielzahl an Fähigkeiten, mit denen er die Umgebung wahrnehmen, interpretieren und mit ihr interagieren kann. Nach einer Studie von McKinsey [McKinsey&Company(2017)] wurden 18 Fähigkeiten in fünf



(a) Pick-by-Vision, xPick von TeamViewer
[TeamViewer(2021)]

(b) Pick-by-Spot, PickVision von WIBOND
[WIBOND(2021)]

Abbildung 2.6: Intelligente Assistenten der menschlichen Kommissionierung

Kategorien identifiziert, die für ein autonom agierendes System funktionsrelevant sind.

Die fünf Kategorien sind hierbei:

1. **Sensorische Fähigkeiten:** Hierzu gehört die visuelle Perzeption, das taktile Fühlen sowie das Hören
2. **Kognitive Fähigkeiten:** Hierzu gehören viele verschiedene Fähigkeiten, wie z.B. das Erkennen von bekannten Mustern, logisches Denken oder Kreativität
3. **Natürliche Sprachverarbeitung** besteht zum einen in der Generierung von Sprachnachrichten und zum anderen im Verständnis von Sprachnachrichten
4. **Soziale und Emotionale Fähigkeiten** besteht zum einen in der Generierung von Sprachnachrichten und zum anderen im Verständnis von Sprachnachrichten
5. **Physische Fähigkeiten** wie Grobmotorik, Feinmotorik, Navigation und Mobilität

Des Weiteren wird in der Studie darauf eingegangen, inwieweit einzelne Teilgebiete bereits automatisiert funktionieren und in einem nächsten Schritt die jeweilige Systemleistung mit der eines Menschen verglichen (Abbildung 2.7).

		Automation capability	Capability level ¹	Description (ability to ...)
Sensory perception	Sensory perception		Yellow	Autonomously infer and integrate complex external perception using sensors
Cognitive capabilities	Recognizing known patterns/categories (supervised learning)		Green	Recognize simple/complex known patterns and categories other than sensory perception
	Generating novel patterns/categories		Red	Create and recognize new patterns/categories (e.g., hypothesized categories)
	Logical reasoning/ problem solving		Red	Solve problems in an organized way using contextual information and increasingly complex input variables other than optimization and planning
	Optimization and planning		Green	Optimize and plan for objective outcomes across various constraints
	Creativity		Red	Create diverse and novel ideas, or novel combinations of ideas
	Information retrieval		Green	Search and retrieve information from a large scale of sources (breadth, depth, and degree of integration)
	Coordination with multiple agents		Red	Interact with others, including humans, to coordinate group activity
	Output articulation/ presentation		Yellow	Deliver outputs/visualizations across a variety of mediums other than natural language
Natural language processing	Natural language generation		Yellow	Deliver messages in natural language, including nuanced human interaction and some quasi language (e.g., gestures)
	Natural language understanding		Red	Comprehend language, including nuanced human interaction
Social and emotional capabilities	Social and emotional sensing		Red	Identify social and emotional state
	Social and emotional reasoning		Red	Accurately draw conclusions about social and emotional state, and determine appropriate response/action
	Social and emotional output		Red	Produce emotionally appropriate output (e.g., speech, body language)
Physical capabilities	Fine motor skills/dexterity		Yellow	Manipulate objects with dexterity and sensitivity
	Gross motor skills		Green	Move objects with multidimensional motor skills
	Navigation		Green	Autonomously navigate in various environments
	Mobility		Red	Move within and across various environments and terrain

Abbildung 2.7: Grad der Automatisierung für die 18 identifizierten Teilbereiche der menschlichen Leistungsfähigkeit [McKinsey&Company(2017)]

Hierbei konnte gezeigt werden, dass insbesondere bei der Generierung von neuen Mustern, der logischen Schlussfolgerung, der Kreativität, der Mobilität und der Koordination von mehreren Agenten ein starkes Defizit im Bereich der kognitiven Fähigkeiten von automatisierten Systemen vorliegt. Bei sozialen und emotionalen Fähigkeiten sind alle genannten Kriterien weit hinter denen des Menschen. Im Bereich physische Fähigkeiten existieren bei Feinmotorik und Geschicklichkeit weitere Verbesserungsmöglichkeiten, für Grobmotorik und

Navigation sind automatisierte Lösungen mit denen eines Menschen durchaus vergleichbar, ebenso bei der Erkennung von bekannten Mustern.

Weiterhin wird die Wichtigkeit der Kombination der einzelnen Punkte hervorgehoben, häufig werden in realen Aufgaben Fähigkeiten aller fünf Kategorien gleichzeitig benötigt. Heutige autonome Systeme beherrschen diese Kombinationsfähigkeit nicht oder nur stark eingeschränkt. Es ist somit kein Wunder, dass der Mensch nach wie vor bei unvorhersehbaren Ereignissen, sich häufig wechselnden Aufgaben und schwer manipulierbaren Gegenständen die Leistungsfähigkeit moderner CPS übertrifft. Der Vorsprung basiert insbesondere auf kombiniertem Modellwissen, wie Sachverhalte zu interpretieren und bestimmte Umgebungsreaktionen auf Basis des eigenen Handelns vorherzusagen sind.

2.4 Teleoperation und Autonomie

Bereits im Jahr 1978 veröffentlichte Sheridan eine Arbeit über die Möglichkeiten der Teleoperation von unbemannten Unterseebooten [Sheridan und Verplank(1978)]. Die dem Computer deutlich überlegenen menschlichen Fähigkeiten sollten bestmöglich an ein CPS übertragen werden. Abbildung 2.8 zeigt seine Kategorisierung von menschlicher Arbeit im Vergleich zu der eines Roboters. Auf der x-Achse wird der *Grad der Automatisierung* (engl. Degree of Automation) von *vollständig manuell* bis *vollständig automatisiert* dargestellt. Die y-Achse zeigt die *Aufgabenentropie* (engl. Task Entropy) von *vordefiniert* bis *unvorhersehbar*.

Der *perfekte Roboter*, mit einer vollständigen Automatisierung bei unvorhersehbaren Aufgaben, befindet sich rechts oben im Schaubild. Links unten definiert Sheridan den *menschlichen Sklaven*, der vollständig vordefinierte Aufgaben manuell erledigt. Spannendster Aspekt der Grafik ist jedoch die Diagonale, die sich von links oben (*Mensch verrichtet würdevolle Arbeit*) über die *überwachende Steuerung* bis hin zu *aktuelle Robotiksysteme* bewegt. Sheridan beschreibt diese Diagonale als aktuellen Forschungsstand der Robotik und sieht die Teleoperation als Möglichkeit, die aktuellen Fähigkeiten der Robotik sinnvoll zu ergänzen.

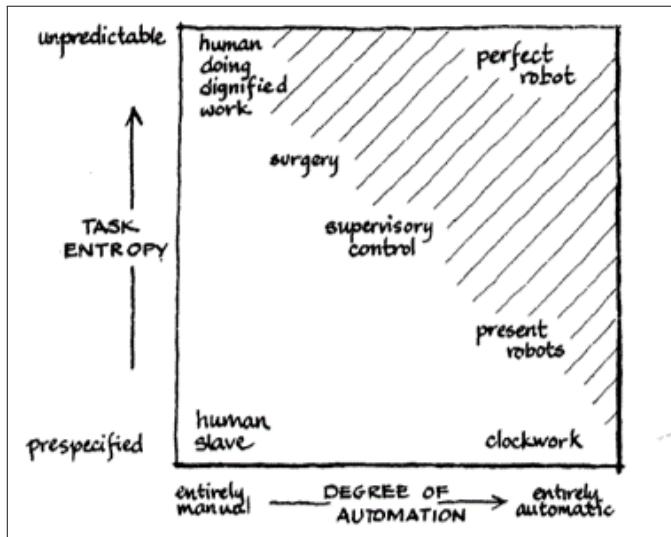


Abbildung 2.8: Klassifikation der Arbeit nach Sheridan und Verplank(1978)]

Abbildung 2.9 zeigt den Vergleich zwischen *Direkter Steuerung* und *Überwachender Steuerung* bei der Teleoperation.

Der Unterschied zwischen den beiden Steuerungsmöglichkeiten besteht vor allem in der Datenverarbeitung bzw. Datenübertragung. Während bei der *direkten Steuerung* die Ausgabegeräte auf Seite des Teleoperators direkt an die Sensoren des CPS angeschlossen werden, ist bei der *überwachenden Steuerung* die Verwendung zweier weiterer Computer notwendig: Einer auf Seite des Teleoperators, der die Ein- und Ausgabegeräte ansteuert und die Datenübertragung zum CPS realisiert und einer auf Seite des CPS, der die Kommunikation zum Teleoperator herstellt und angeschlossene Sensoren und Aktoren ausliest bzw. steuert.

Die Verwendung eines Computers auf Seite des CPS erlaubt zum einen eine mögliche Zeitverzögerung zwischen Signalaufnahme und Steuerung und ermöglicht zudem eine automatisierte Steuerung des Systems. Die Fähigkeit Aufgaben selbstständig zu lösen, wird für computergesteuerte Systeme durch verschiedene Begrifflichkeiten und in verschiedenen *Stufen der Automatisierung*

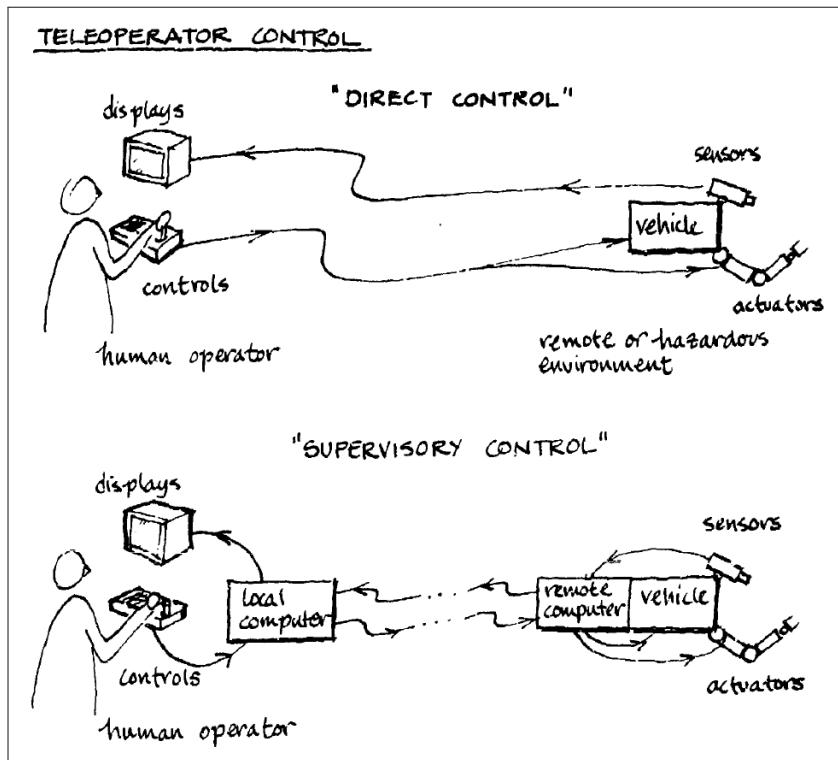


Abbildung 2.9: Unterschied zwischen *Direkter Steuerung* und *Überwachender Steuerung* bei der Teleoperation nach Sheridan und Verplank(1978)]

rung (engl. Levels of Automation - LOA) beschrieben, die häufig im Kontext mit selbstfahrenden Personenkraftwagen (PKWs) Verwendung finden.

In der Literatur findet sich eine Vielzahl an Definitionen und Beschreibungen zu den Stufen der Automatisierung und der Interaktion zwischen Mensch und Maschine [Sheridan und Verplank(1978)], [Parasuraman et al.(2000)Parasuraman, Sheridan, und Wickens], [Trafton und Schultz et al.(2004)], [Frohm et al.(2008)Frohm, Lindström, und Winroth et al.], [Sheridan(2011)], [Save und Feuerberg(2014)], [Beer et al.(2014)Beer, Fisk, und Rogers]. Inhaltlich unterscheiden sich diese entweder in der Aufteilung der Stufen, im Fokus auf Autonomie oder manuelle

Arbeit, im Anwendungsfeld oder im Grad der Abstraktion bezüglich Funktionalität und Informationsaustausch. Nachfolgend sind die zehn Stufen nach Sheridan [Sheridan und Verplank(1978)] aufgeführt, die das grundsätzliche Konstrukt gesamtheitlich beschreiben:

Stufe	Beschreibung
1	Der Computer stellt keine Assistenz bereit, der Mensch übernimmt jegliche Entscheidungen und Aktionen
2	Der Computer stellt die vollständige Auswahl an möglichen Entscheidungen / Aktionen zur Verfügung, oder grenzt die Auswahl auf einige wenige ein, oder schlägt eine Alternative vor
5	führt den Vorschlag aus, nachdem der Mensch bestätigt hat erlaubt dem Menschen eine bestimmte Zeit ein Veto einzulegen
6	bevor die Aktion automatisch ausgeführt wird
7	führt die Aufgabe automatisch aus und informiert den Menschen
8	führt die Aufgabe automatisch aus, informiert nur auf Anfrage, oder
9	informiert den Menschen nur, wenn sich das System dazu entscheidet
10	Das System entscheidet alles, agiert autonom, Mensch wird ignoriert

Tabelle 2.1: Die 10 Stufen der Autonomie, adaptiert aus Sheridan [Sheridan und Verplank(1978)]

Die Stufe der Autonomie ist stets gegenläufig zum Grad der Interaktion mit dem Menschen. In welchen Umfang einem CPS assistiert werden kann, ist somit bereits durch die jeweilige Stufe der Automatisierung definiert. Beer [Beer et al.(2014)Beer, Fisk, und Rogers] ergänzt die Stufen der Automatisierung um den Bezug der Mensch-Roboter-Interaktion durch drei Kriterien, die aufgabenspezifisch die Autonomie des Systems beschreiben sollen: *erkennen*, *planen* und *handeln*. Eine qualitative Bewertung der drei Kriterien führt zu der aktuellen Autonomie des Systems bzw. zur notwendigen Hilfestellung des Menschen.

Mit der Einführung der *Oculus Rift* im Jahr 2013 [DJSCOE, Vile - Parle (W), Mumbai und Rajesh Desai et al.(2014)], schaffen es Virtual-Reality Systeme erstmals, die visuellen Anforderungen [Gerschütz et al.(2019)Gerschütz, Fechter, Schleich, und Wartzack] des Menschen zu erfüllen ohne Schwindel oder

Übelkeit hervorzurufen. Seither haben sich VR-Brillen kontinuierlich weiterentwickelt und heben die Immersion eines Menschen innerhalb computergenerierter Umgebungen auf ein neues Niveau [Angelov et al.(2020)Angelov, Petkov, Shipkovenski, und Kalushkov].

Etliche Forschungsarbeiten untersuchen Telepräsenz und Teleoperation in Kombination mit der virtuellen Realität [Geiger et al.(2010)Geiger, Popp, Färber, Artigas, und Kremer], [Peppoloni et al.(2015)Peppoloni, Brizzi, Rufaldì, und Avizzano], [Kent et al.(2017)Kent, Saldanha, und Chernova], [Brizzi et al.(2018)Brizzi, Peppoloni, und Graziano et al.], [Stotko und Krumpen et al.(2019)], [Kent et al.(2020)Kent, Saldanha, und Chernova]. Die Ergebnisse favorisieren stets die Visualisierung und Steuerung innerhalb einer virtuellen Umgebung. Die am häufigsten auftretenden Probleme sind Latenz, Datenmenge und physische Limitierungen der eingesetzten Robotik oder mechanische und steuerungstechnische Probleme bei der Verwendung von bewegten Kameras [Theofilis et al.(2016)Theofilis, Orlosky, Nagai, und Kiyokawa].

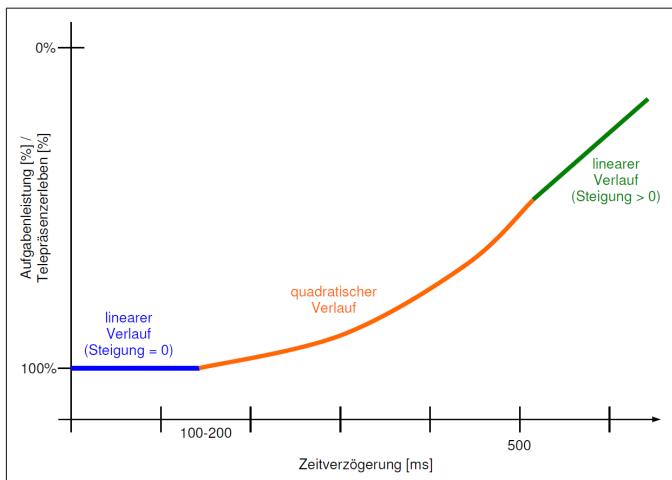


Abbildung 2.10: Angenommener Verlauf der Leistungsverschlechterung bzw. des Telepräsenzerlebens in Abhängigkeit von der Zeitverzögerung (visuell bzw. visuell/haptisch). Modell basiert auf der Auswertung mehrerer Studien zum Thema Teleoperation unter Latenz [Pongrac(2008)]

Die Arbeit von Pongrac [Pongrac(2008)] bietet eine hervorragende Zusammenstellung durchgeführter Versuche zur Untersuchung der Teleoperation bei wirkender Latenz. Pongrac erstellte auf Basis der gewonnenen Daten aus mehreren Studien ein hypothetisches Modell, das mit einer Kombination aus linearer und quadratischer Funktion den Verlauf der Aufgabenleistung in Abhängigkeit der Latenz abbildet (Abbildung 2.10). Das Modell beinhaltet Studienergebnisse von teleoperierten Greif- und Platzieraufgaben.

Pongrac unterscheidet in ihrer Arbeit zwischen zwei unterschiedlichen Aufgabentypen: Freie Bewegung (z.B. Greifen und Ablegen eines Objekts) und Kontaktaufgaben (z.B. Öffnen eines Bajonett-Verschlusses). Weiterhin wird zwischen Aufgabenleistung und Präsenzempfinden bei der Teleoperation unterschieden. Während sich nach ihrem Modell die Aufgabenleistung erst für Latenzen ab $200ms$ messbar verschlechtern, verringert sich das subjektive Präsenzempfinden bereits ab $150ms$. Anschließend verlaufen beide Kriterien quadratisch bis zu einer Latenz von $400ms$ für das Präsenzempfinden bzw. $500ms$ für die Aufgabenleistung. Für Latenzen über $500ms$ stellt sich ein linear steigender Verlauf der Leistungsverschlechterung ein.



Abbildung 2.11: VR-Teleoperation eines Knickarmroboters für freie Bewegungen (Greifen des silbernen Griffes) oder Kontaktaufgaben (Öffnen eines Bajonett-Verschlusses) [Pongrac(2008)]

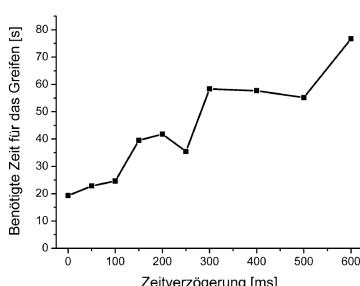
Begründet wird der Wechsel zu einem linearen Verlauf ab $500ms$ durch das Wegfallen eines kognitiven Abgleichs zwischen visueller und haptischer Modalität des Menschen [Pongrac(2008)]. Entsprechend entwickelt der Mensch eine *Move-Wait-Strategie*, deren weitere Reduzierung der Aufgabenleistung lediglich linear von der zusätzlich wirkenden Latenz abhängt.

Die Versuchsdurchführung für das teleoperierte Greifen wurde mit 30 Proband*innen und jeweils einer Iteration bei diversen Latenzen zwischen $0ms$ und $600ms$ durchgeführt. Die Proband*innen mussten den Parallelgreifer eines Knickarmroboters zu einem Bügel bewegen und diesen greifen. Eine weitere Bewegung wurde nicht ausgeführt. Der Versuchsaufbau ist in Abbildung 2.11 dargestellt.

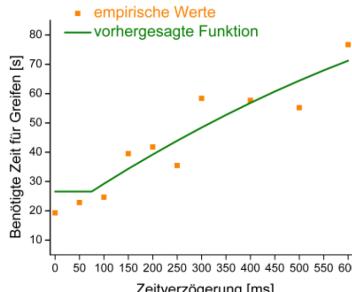
Die benötigte Zeit für das Greifen T in Abhängigkeit der Latenz L sowie die Anwendung des entwickelten Modells für $T(L)$ mit der Formel

$$T(L) = \begin{cases} 26.565 & L \leq 75ms \\ 18.435 + 0.112L - 4 \cdot 10^{-5}L^2 & 75ms < L \leq 600ms \end{cases} \quad (2.1)$$

ist in Abbildung 2.12 dargestellt.



(a) Messwerte



(b) Regression

Abbildung 2.12: Ergebnisse von Pongrac [Pongrac(2008)] für das teleoperierte Greifen (freie Teleoperation) unter Latenz

Die Auswertung von Pongrac geht weiterhin von einem linearen Verlauf mit einer Steigung von 0 für Latenzen $< 75ms$ aus. Anschließend wird von einem quadratischen Verlauf ausgegangen.

Schwerpunkt der Arbeit von Zhang [Zhang et al.(2018)] ist nicht nur die Steuerung eines Roboters in der virtuellen Realität, sondern das Ableiten von Regeln aus dem menschlichen

Verhalten. Abbildung 2.13 zeigt einen Menschen bei der VR-Teleoperation eines Serviceroboters.

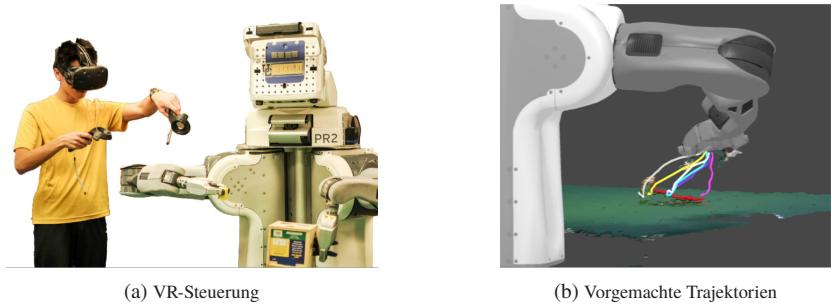


Abbildung 2.13: VR-Teleoperation eines Knickarmroboters mit vorgemachten Trajektorien zur Nachahmung durch den Roboter mit Hilfe eines Neuronalen Netzes [Zhang et al.(2018)Zhang, McCarthy, Jow, Lee, Chen, Goldberg, und Abbeel]

Das zugehörige Forschungsfeld nennt sich *Imitation Learning* und beschreibt die Wissenschaft, wie ein System aus den Aktionen eines anderen Systems (auch als Expertensystem bezeichnet) Regeln oder Modelle ableiten kann. Anschließend soll es bestmöglich die vorgemachten Aktionen nachahmen. Nach [Osa et al.(2018)Osa, Pajarinen, Neumann, Bagnell, Abbeel, und Peters] können die Bilder einer Kamera, Gelenkwinkel eines Roboters oder Bewegungen eines Teleoperators als Eingangsdaten fungieren. Das Verhalten des nachahmenden Systems wird meist durch tiefe neuronale Netze abgebildet und auf Basis eines großen Datensatzes an Expertendemonstrationen trainiert.

Ansätze, die nur wenige Demonstrationen benötigen, wie das von Duan vorgestellte *Oneshot Imitation Learning* [Duan und Andrychowicz et al.(2017)], werden Umgebung bzw. die zu handhabenden Objekte in einem großen Datensatz vortrainiert. Das angeeignete Modellwissen erlaubt eine schnelle Nachahmung bei lediglich einer Demonstration mit den bereits bekannten Objekten. Einen ähnlichen Ansatz verfolgt James mit *Task-Embedded Control Networks for Few-Shot Imitation Learning* [James et al.(2018)James, Bloesch, und Davison], das in einer Simulation vortrainiert werden kann und entsprechend bereits Gelerntes mit wenigen Demonstrationen verknüpft.

2.5 Fazit und Forschungslücke

Wie bereits diskutiert liegen die menschlichen Fähigkeiten nach wie vor weit über denen eines cyber-physischen Systems innerhalb dynamischer Umgebungen und komplexer Aufgabenbereiche.

Während die Grundüberlegungen von Sheridan [Sheridan und Verplank(1978)] über die zehn Stufen der Autonomie von CPS und zwei verschiedene Möglichkeiten der Teleoperation (direkte Steuerung und überwachte Steuerung) eine erste Verknüpfung von maschinellen und menschlichen Fähigkeit darstellt, ist unklar, wie eine allgemeingültige Formulierung der Kooperation aussehen sollte. Auch die beschriebenen Definitionen von Beer [Beer et al.(2014)Beer, Fisk, und Rogers] adressieren nicht die Art der Teleoperation unter Berücksichtigung bzw. Nutzung der physischen und kognitiven Fähigkeiten des Menschen. Inwiefern durch diese Kopplung eine nachhaltige Leistungssteigerung für CPS durch eine Teleoperation aussehen kann, bleibt unerwähnt.

Im Rahmen der vorliegenden Arbeit sollen die Grundüberlegungen von Sheridan und Beer übernommen, deren Umsetzung technologisch auf den neusten Stand gebracht und in ein gültiges Framework des kontinuierlichen Lernens für cyber-physischen Systeme durch Teleoperation überführt werden.

Wie eine Einbindung des Menschen hierbei zu realisieren ist und welche Module für ein solches Framework notwendig sind, soll mit der ersten forschungsleitenden Fragestellung beantwortet werden:

Fragestellung 1: Wie kann die menschliche Überlegenheit bei der Dateninterpretation in komplexe Prozessabläufe von cyber-physischen Systemen integriert werden?

Eine Kategorisierung der Hilfestellung des Menschen durch Teleoperation, mit direktem Bezug zu den eingesetzten menschlichen Fähigkeiten und eingesetzter Technologie des CPS, findet sich zum heutigen Tage nicht in der Literatur und soll durch die zweite Fragestellung beantwortet werden:

Fragestellung 2: In welche Stufen lässt sich Teleoperation für cyber-physische Systeme kategorisieren?

Das entwickelte Framework soll am Beispiel eines Kommissionierroboters implementiert und validiert werden. Notwendige Module und Algorithmen werden durch die dritte forschungsleitende Fragestellung untersucht:

Fragestellung 3: Wie lässt sich eine konkrete Umsetzung eines solchen Systems realisieren und welche Module sind dafür notwendig?

Zwei große Hemmnisse für den Einsatz von moderner Automatisierungstechnik konnten im deutschen Mittelstand identifiziert werden [accenture und Welt(2020)]: Mangelnde Fachkräfte und eine hohe Störanfälligkeit der Systeme.

Die vierte forschungsleitende Fragestellung adressiert die Problematik mangelnder Fachkräfte. Insbesondere Personen die wenig bis keine Erfahrung im Bereich Robotik haben, sollen das entwickelte System innerhalb eines realen Szenarios teleoperieren:

Fragestellung 4: Kann eine Hilfestellung des Menschen via Teleoperation bei komplexen Aufgabenstellungen benutzerfreundlich und intuitiv gestaltet werden?

Die Arbeit von Pongrac [Pongrac(2008)] untersucht ausführlich den Einfluss von Latenz auf die Teleoperation. Inwiefern sich die Ergebnisse für einfache Greifaufgaben (nur Anfahrt des Roboters ohne Ablegen) auf einen vollständigen Kommissionierprozess übertragen lassen, ist bisher nicht untersucht. Ebenso ist fraglich, ob das entwickelte Modell mit einer Kombination aus linearem und quadratischem Verlauf weiterhin gültig ist.

Die fünfte forschungsleitende Fragestellung soll mit einer ausführliche Proband*innenstudie Auskunft über den Einfluss von Latenz während der Teleoperation von Kommissionierrobotern, unter Einsatz neuster technologischer Entwicklungen, geben:

Fragestellung 5: Wie ist der Einfluss von Latenz zwischen Teleoperator und CPS?

Die noch offene Störanfälligkeit soll gemeinsam mit der Frage über die Lernfähigkeit von CPS durch Teleoperation beantwortet werden.

Die vorgestellten Verfahren [Duan und Andrychowicz et al.(2017)], [James et al.(2018)James, Bloesch, und Davison], [Zhang et al.(2018)Zhang, McCar-

thy, Jow, Lee, Chen, Goldberg, und Abbeel] benötigen große Datenmengen über das exakte Einsatzszenario des Systems. Häufig sind es jedoch unerwartete Situationen, die ein CPS zum Ausfall bringen. Die Frage, ob bei neuen Situationen durch Teleoperation die Leistungsfähigkeit eines Systems unmittelbar gesteigert werden kann und diese, im Falle eines erneuten Auftretens, bereits automatisiert gelöst werden können, soll mit der sechsten und letzten forschungsleitenden Fragestellung untersucht werden:

Fragestellung 6: Kann durch das entwickelte Framework die Leistungsfähigkeit eines CPS mittelbar und unmittelbar verbessert werden?

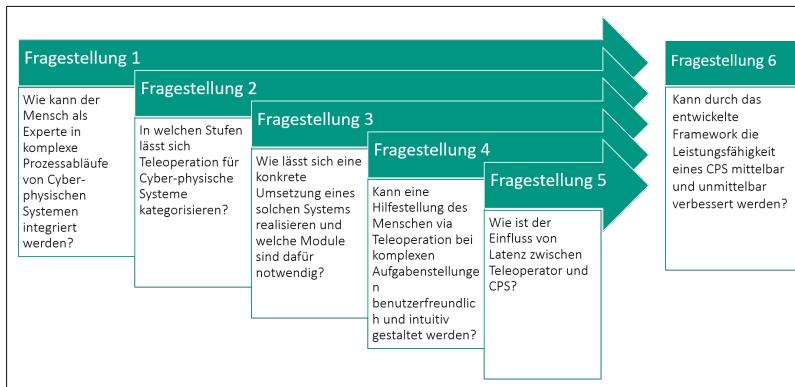


Abbildung 2.14: Übersicht der forschungsleitenden Fragestellungen, eigene Darstellung

Zur Beantwortung der letzten forschungsleitenden Fragestellung fließen die Entwicklungen und Erkenntnisse aller vorhergehenden Fragestellungen ein. Die Beantwortung kann somit als Funktionsdemonstration des entwickelten Frameworks verstanden werden (Abbildung 2.14).

3 Stand der Technik

3.1 Mathematische Grundlagen

Innerhalb der Robotik finden folgende Begriffsdefinitionen Verwendung [Mareczek(2020)]:

Gelenke Gelenke verbinden zwei aufeinander folgende Glieder einer kinematischen Kette und schränken deren Relativbewegung auf die Freiheitsgrade des Gelenks ein.

Generalisierte Koordinaten Generalisierte Koordinaten beschreiben den minimalen Satz an Parametern, die für eine eindeutige Beschreibung im Raum benötigt werden.

Gelenkraum Der Gelenkraum beschreibt mit Hilfe der generalisierten Koordinaten die eindeutige Position des Knickarmroboters. Die Gelenke können hierbei entweder einen rotatorischen oder einen translatorischen Freiheitsgrad besitzen.

Endeffektor Der Endeffektor wird als das letzte Element der offenen kinematischen Kette des Knickarmroboters bezeichnet. Im Falle eines Robotiksystems wird z.B. der Greifer (Effektor) als Endeffektor bezeichnet.

Konfiguration Die Konfiguration eines Knickarmroboters beschreibt im Falle mehrerer möglicher Gelenkwinkelkombinationen für eine Endeffektorposition die Stellung der verantwortlichen Gelenke.

Pose Eine Pose beschreibt die Position und Orientierung des Endeffektors in sechs Dimensionen im kartesischen Raum, bzw. in sieben Dimensionen, sollte die Orientierung in Quaternionen (siehe Kapitel 3.1.2) formuliert sein.

Vollständiger Arbeitsraum Der vollständige Arbeitsraum eines Knickarm-roboters bezeichnet den Raum, der vom Endeffektor ohne Vorgabe der Orientierung des Endeffektors erreichbar ist.

Erreichbarer Arbeitsraum Der erreichbare Arbeitsraum schränkt den vollständigen Arbeitsraum durch eine Vorgabe der Endeffektororientierung ein.

Singularität Eine Singularität entsteht, wenn für eine Pose unendlich viele Lösungen innerhalb des Gelenkkraums entstehen.

3.1.1 Koordinatensysteme

Die Position eines Punktes im dreidimensionalen Raum kann durch verschiedene Koordinatensysteme beschrieben werden:

- Kartesische Koordinaten
- Kugelkoordinaten

Kartesische Koordinaten werden durch drei zueinander rechtwinklig angeordnete Achsen beschrieben, die sich im Koordinatenursprung O schneiden. Der Raum wird somit durch ein rechteckiges Koordinatensystem beschrieben.

Kugelkoordinaten beschreiben die Position im Raum durch einen Radius r und die beiden Winkel Θ und ϕ .

In Abbildung 3.1 wird die Position des Punktes P in kartesischen Koordinaten und in Kugelkoordinaten beschrieben.

Die Umrechnung zwischen beiden Koordinatensystemen erfolgt nach folgender Vorschrift:

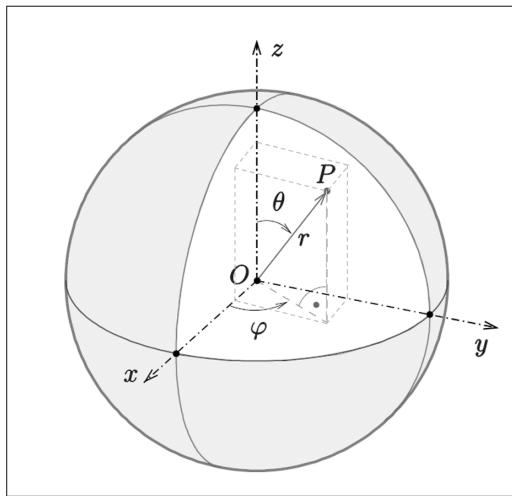


Abbildung 3.1: Vergleich von kartesischen Koordinaten und Kugelkoordinaten, Abbildung adaptiert von [Wikimedia(2020a)]

$$\begin{aligned}
 x &= r \cdot s\theta \cdot c\varphi \\
 y &= r \cdot s\theta \cdot s\varphi \\
 z &= r \cdot c\theta
 \end{aligned} \tag{3.1}$$

sowie in umgekehrte Richtung:

$$\begin{aligned}
 r &= \sqrt{x^2 + y^2 + z^2} \\
 \theta &= \arccos \frac{z}{r} \\
 \varphi &= \arctan2(y, x)
 \end{aligned} \tag{3.2}$$

Soweit nicht weiter ausgeführt ist im Folgenden von der Verwendung des kartesischen Koordinatensystems auszugehen.

3.1.2 Rotation im Raum

Nach [Rill et al.(2020)] wird die relative Position eines Objektes im dreidimensionalen Raum durch den Vektor $\vec{p} = (x, y, z)$ beschrieben. Die Orientierung kann entweder durch Euler-Winkel, Kardanwinkel oder Quaternionen beschrieben werden (eigene Darstellung). Im folgenden wird von einem raumfesten Koordinatensystem K_0 und einem körperfesten Koordinatensystem K_f ausgegangen.

Euler-Winkel

Euler-Winkel beschreiben die Orientierung eines Objektes im dreidimensionalen euklidischen Raum. Die Umrechnung zwischen zwei rotierten Koordinatensystemen wird mit Hilfe einer Rotationsmatrix durchgeführt. Die Rotationsmatrix R ist eine 3×3 Matrix der Form

$$R \triangleq \begin{bmatrix} r_{11} & r_{12} & \dots & r_{1n} \\ r_{21} & r_{22} & \dots & r_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ r_{m1} & r_{m2} & \dots & r_{mn} \end{bmatrix} \quad (3.3)$$

und wird aus den drei Elementardrehungen bestimmt:

$$R_x(\alpha) = \begin{bmatrix} c\alpha & -s\alpha & 0 \\ s\alpha & c\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.4)$$

$$R_y(\beta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c\beta & -s\beta \\ 0 & s\beta & c\beta \end{bmatrix} \quad (3.5)$$

$$R_z(\gamma) = \begin{bmatrix} c\gamma & -s\gamma & 0 \\ s\gamma & c\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.6)$$

Die erste und letzte Rotation findet bei eigentlichen Euler-Winkeln um die gleiche Achse statt. Die Rotationsmatrix wird durch Matrixmultiplikation der Elementardrehungen bestimmt. Gleichung 3.7 zeigt die ausmultiplizierte Transformationsmatrix nach zxz -Drehung.

$$R_{zxz} = R_z(\alpha) \circ R_x(\beta) \circ R_z(\gamma) =$$

$$\begin{bmatrix} c\alpha c\gamma - s\alpha c\beta s\gamma & -c\alpha s\gamma - s\alpha c\beta c\gamma & s\alpha s\beta \\ s\alpha c\gamma + c\alpha c\beta s\gamma & -s\alpha s\gamma + c\alpha c\beta c\gamma & -c\alpha s\beta \\ s\beta s\gamma & s\beta c\gamma & c\beta \end{bmatrix} \quad (3.7)$$

Eine Transformation des Punktes \vec{p}_{K_k} aus dem körperfesten Koordinatensystem K_k in das raumfeste Koordinatensystem K_0 wird nach folgender Vorschrift durchgeführt:

$$\vec{p}_{K_0} = R_{zxz} \begin{pmatrix} p_x \\ p_y \\ p_z \end{pmatrix}_{K_k} \quad (3.8)$$

Die Transformation in die entgegengesetzte Richtung kann mit der inversen Transformationsmatrix R_{zxz}^{-1} durchgeführt werden.

$$\vec{p}_{K_k} = R_{zxz}^{-1} \begin{pmatrix} p_x \\ p_y \\ p_z \end{pmatrix}_{K_0} \quad (3.9)$$

Roll-Nick-Gier-Winkel

Die Roll-Nick-Gier-Winkel (engl. Roll-Pitch-Yaw angles) sind eine spezielle Variante der Euler-Winkel. Hierbei findet die Rotation durch Drehung um drei verschiedene Koordinatenachsen, anstelle von zwei, statt [Rill et al.(2020)Rill, Schaeffer, und Borchsenius]. Die zu verwendenden Achsen werden durch das körperfeste Koordinatensystem vorgegeben und erlauben eine intuitive Beschreibung der Orientierung des Objektes.

Homogene Koordinatentransformation

Die homogene Koordinatentransformation stellt eine praktische Möglichkeit dar, eine relative Rotation und eine relative Verschiebung zweier Koordinatensysteme innerhalb einer Matrix zu beschreiben. Sie kombiniert die Rotationsmatrix $R \in \mathbb{R}^{3x3}$ mit dem Translationsvektor $\vec{p} \in \mathbb{R}^3$ in der homogenen Transformationsmatrix $T \in \mathbb{R}^{4x4}$:

$$T = \begin{bmatrix} R & p \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & p_1 \\ r_{21} & r_{22} & r_{23} & p_2 \\ r_{31} & r_{32} & r_{33} & p_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.10)$$

Die homogene Transformationsmatrix ermöglicht nach [Lynch und Park(2017)] folgende drei Anwendungsfälle:

- (1) Beschreibung der Pose eines Objekts
- (2) Wechsel des Referenz-Koordinatensystems, in dem ein Objekt beschrieben wird
- (3) Verschiebung und Rotation eines Objekts oder Vektors

Die Beschreibung der Pose eines Objektes (1) wird bereits von der Transformationsmatrix selbst beschrieben.

Der Wechsel des Referenz-Koordinatensystems (2) lässt sich durch die Multiplikation zweier Transformationsmatrizen realisieren

$$T_{ac} = T_{ab}T_{bc} \quad (3.11)$$

wobei T_{ab} die Pose von b in a beschreibt und T_{bc} die Pose von c in b beschreibt. Resultierend beschreibt T_{ac} die Pose von c in a .

Die Verschiebung und Rotation eines Objekts oder Vektors (3) lässt sich durch eine Vektor-Matrix-Multiplikation realisieren:

$$\vec{p}' = T\vec{p} \quad (3.12)$$

Quaternionen

Die Verwendung von Euler-Winkeln erlaubt eine intuitive Darstellung von Rotationen im dreidimensionalen euklidischen Raum. Liegen jedoch zwei Rotationsachsen in einer Ebene, wird dieser Zustand als Singularität oder auch als Gimbal-Lock bezeichnet [Huang(2017)] und führt zum Verlust eines Freiheitsgrades.

Quaternionen sind eine Erweiterung der komplexen Zahlen und ihre Anwendung ermöglicht eine Rotation im dreidimensionalen Raum ohne Singularität. Quaternionen sind nach [Gattringer(2011)] vierdimensionale hyperkomplexe Zahlen und werden durch einen Realteil $s \in \mathbb{R}$ und drei imaginäre Komponenten $q_1, q_2, q_3 \in \mathbb{C}$ dargestellt. Der Aufbau ist wie folgt:

$$\mathbf{q} = s + q_1 \mathbf{i} + q_2 \mathbf{j} + q_3 \mathbf{k}, \quad s, q_1, q_2, q_3 \in \mathbb{R} \\ i, j, k \in \mathbb{C}$$

Wie bei den komplexen Zahlen gilt $i^2 = j^2 = k^2 = -1$. Allerdings ist die Multiplikation von Quaternionen nicht kommutativ. Es gilt:

$$\begin{aligned} i \cdot j &= k & j \cdot k &= i & k \cdot i &= j \\ j \cdot i &= -k & k \cdot j &= -i & i \cdot k &= -j \end{aligned} \quad (3.13)$$

Mit dem Einheitsquaternions $\hat{\mathbf{q}}$, also einem Quaternionen dessen Betrag eins ergibt, können eindeutige, dreidimensionale Rotationen berechnet werden. Der Betrag berechnet sich analog zu Vektoren im \mathbb{R}^3

$$\|\mathbf{q}\| = \sqrt{s^2 + q_1^2 + q_2^2 + q_3^2} \quad (3.14)$$

und ermöglicht die Normierung beliebiger Quaternionen:

$$\hat{\mathbf{q}} = \frac{\mathbf{q}}{\sqrt{s^2 + q_1^2 + q_2^2 + q_3^2}} \quad (3.15)$$

Die Rotation wird nun durch den Winkel ϕ um eine vorgegebene raumfeste Drehachse \vec{u} beschrieben, die sowohl im raumfesten Koordinatensystem K_0 als auch im körperfesten Koordinatensystem K_f gleich ist [Wittenburg et al.(2014)Wittenburg, Richard, Zierep, und Bühler]. Die vier benötigten Parameter werden auch Euler-Parameter genannt [Woernle(2011)] und nach folgender Vorschrift bestimmt:

$$\begin{aligned} s &= c \frac{\phi}{2} & q_1 &= u_1 s \frac{\phi}{2} \\ q_2 &= u_2 s \frac{\phi}{2} & q_3 &= u_3 s \frac{\phi}{2} \end{aligned} \quad (3.16)$$

Eine Rotation eines Vektors \vec{p}_{K_0} im raumfesten Koordinatensystem um den Winkel ϕ wird schließlich mit Hilfe des doppelten Quaternionenprodukts in das körperfeste Koordinatensystem überführt:

$$\vec{p}_{K_k} = q_\phi \circ \vec{p}_{K_0} \circ \overline{q_\phi} \quad (3.17)$$

Eine Koordinatentransformation vom körperfesten Koordinatensystem K_k in das raumfeste Koordinatensystem K_0 kann analog mit Hilfe des doppelten Quaternionenprodukts und dem Transformationsquaternion \mathbf{q}_{k0} bestimmt werden.

3.1.3 Roboterkinematik

Direkte Kinematik

Die Denavit-Hartenberg Parameter (DH-Parameter), auch DH-Transformation oder DH-Konvention genannt, ist ein mathematisches Verfahren, um den Zusammenhang kinematischer Ketten zu beschreiben und wendet die Transformationsvorschrift nach Euler-Winkeln (siehe Kapitel 3.1.2) an. Nach [Lynch und Park(2017)] wird ausgehend von einer offenen kinematischen Kette die Verbindung von i Gelenken durch i Transformationsvorschriften $T_{i-1,i}(\theta)$ beschrieben, wobei θ den zugehörigen Gelenkwinkel darstellt. Die abzubildende Kette ist hierbei nicht auf rotatorische Freiheitsgrade limitiert, sondern kann auch prismatische Gelenke (Schubgelenke) abbilden.

Jede Transformationsvorschrift besteht aus lediglich vier Parametern:

- d : Distanz zwischen vorheriger und aktueller x-Achse, entlang der vorherigen z-Achse
- ϕ : Winkel um die z-Achse zwischen vorheriger und aktueller x-Achse
- a : Länge der gemeinsamen Normalen, Abstand zwischen vorheriger und aktueller z-Achse
- α : Winkel um die gemeinsame Normale zwischen vorheriger und aktueller z-Achse

Nachfolgend wird von der Beschreibung der modifizierten DH-Parameter ausgegangen. Die Transformationsvorschrift von Koordinatensystem K_{i-1} zu K_i wird durch die homogene Transformationsmatrix (siehe Kapitel 3.1.2) dargestellt:

$$T_{i-1,i} = \begin{bmatrix} c\phi_i & -s\phi_i & 0 & a_{i-1} \\ s\phi_i c\alpha_{i-1} & c\phi_i c\alpha_{i-1} & -s\alpha_{i-1} & -d_1 s\alpha_{i-1} \\ s\phi_i s\alpha_{i-1} & c\phi_i s\alpha_{i-1} & c\alpha_{i-1} & d_i c\alpha_{i-1} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.18)$$

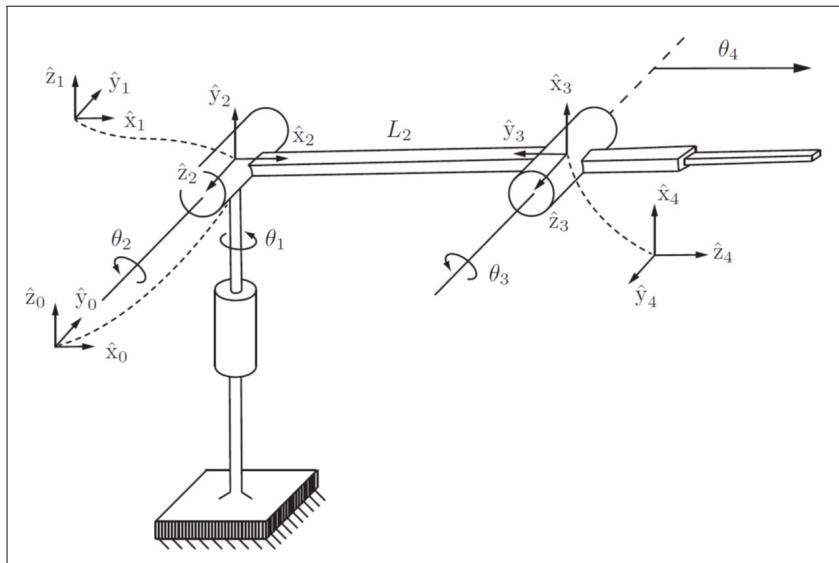


Abbildung 3.2: Beispiel über Koordinatensystemdefinition an einem Knickarmroboter mit vier Freiheitsgraden nach DH-Konvention [Lynch und Park(2017)]

In Abbildung 3.2 ist ein RRRP-Roboter dargestellt, also ein Knickarmroboter mit drei rotatorischen und einem prismatischen Gelenk. Die zugehörigen Koordinatensysteme werden entsprechend der DH-Konvention definiert und anschließend beschrieben. Die angegebenen Winkel θ_{1-3} beschreiben hierbei die jeweilige Rotation, θ_4 beschreibt die Verschiebung des prismatischen Gelenks.

i	α_{i-1}	a_{i-1}	d_i	ϕ_i
1	0°	0	0	θ_1
2	90°	0	0	θ_2
3	0°	0	L_2	$\theta_3 + 90^\circ$
4	90°	0	0	0

Die Koordinatentransformation von K_0 zu K_4 , also die direkte Kinematik oder Vorwärtskinematik von der Roboterbasis zum Endeffektor, lässt sich durch einfache Multiplikation alle Transformationsmatrizen $T_{i-1,i}$ mit $i \in [1, 4]$ bestimmen:

$$T_{0,4}(\theta_1, \theta_2, \theta_3, \theta_4) = T_{0,1}(\theta_1)T_{1,2}(\theta_2)T_{2,3}(\theta_3)T_{3,4}(\theta_4) \quad (3.19)$$

3.2 Netzwerkkommunikation

Das folgende Kapitel beschreibt die Grundlagen der Netzwerkkommunikation. Hierbei soll ausschließlich auf die Kommunikation innerhalb eines lokalen Computernetzwerkes (engl. local area network), auch LAN genannt, eingegangen werden. Die Verwendung eines lokalen Computernetzwerkes gleicht dem einer *Edge-Cloud*, z.B. ein Firmennetzwerk innerhalb der Logistik eines Werkes.

Die Netzwerke teilnehmenden können hierbei entweder über eine Kabelverbindung oder drahtlos über WLAN (engl. wireless lan) mit anderen Netzwerkteilnehmenden kommunizieren.

3.2.1 Netzwerkschichten

Das TCP/IP-Referenzmodell besteht aus vier aufeinander aufbauenden Schichten, die eine reibungslose Kommunikation zwischen zwei Netzwerke teilnehmenden gewährleisten. Eine Definition für die konkrete Umsetzung wurde nicht vorgegeben, aus diesem Grund sind mehrere Protokolle für die jeweiligen Schichten entwickelt worden.

- 1 **Bitübertragungsschicht** Die Bitübertragungsschicht beschreibt die physische Anbindung an das Übertragungsmedium und überträgt binäre Signale (Einsen und Nullen).
- 2 **Sicherungsschicht** Die Sicherungsschicht beinhaltet verschiedene Mechanismen zur Sicherstellung einer

Schicht	Bezeichnung	Protokoll und Aufgabe
1	Netzzugang	Ethernet, WLAN, etc.
1.1*	Bitübertragungsschicht	Baudrate, etc.
1.2*	Sicherungsschicht	Frameerstellung
2	Vermittlungsschicht	IPv4, IPv6, etc.
3	Transportschicht	TCP, UDP
4	Anwendungsschicht	HTTP, FTP, etc.

Tabelle 3.1: Die vier Schichten des TCP/IP-Referenzmodell, sowie mit (*) markierte Einträge aus dem hybriden Referenzmodell, adaptiert aus [Baun(2020)]

erfolgreichen Datenübertragung. Hierzu werden z.B. Prüfsummen gebildet und die Pakete der Nutzschicht in Datenframes eingebettet.

- 3 **Vermittlungsschicht** Die Vermittlungsschicht kümmert sich um das Routing einzelner Pakete innerhalb eines Netzwerkes. Meist wird hier das IP-Protokol (internet protocol) eingesetzt und die Segmente der Transportschicht in Pakete eingebettet.
- 4 **Transportschicht** Die Transportschicht stellt sicher, dass die Daten der Vermittlungsschicht an die zugehörigen Prozesse gesendet werden. Hier kommen die beiden bekanntesten Protokolle TCP oder UDP zum Einsatz.
- 5 **Anwendungsschicht** Die Anwendungsschicht enthält die eigentlichen Daten der Netzwerkkübertragung. Hier können eigene Protokolle definiert oder vorhandene (wie z.B. HTTP, FTP, etc.) genutzt werden.

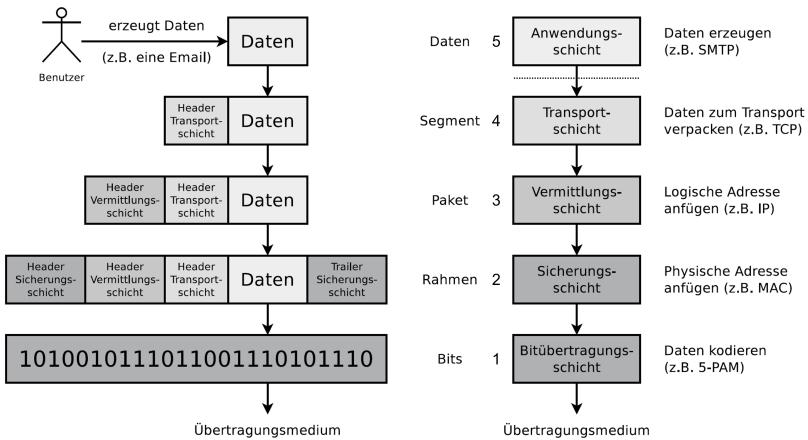


Abbildung 3.3: Datenkapselung durch die verschiedenen Netzwerkschichten des TCP/IP (ggf. hybriden) Referenzmodells nach [Baun(2020)]

3.2.2 Definitionen und Begrifflichkeiten

Folgende Definitionen nach [Wendzel(2018)] und [Baun(2020)] sollen beim Verständnis der Netzwerkkommunikation helfen und die verwendeten Fachbegriffe einführen und erklären.

Ethernet Ethernet (IEEE 802.3) ist eine Technologie aus dem Jahre 1970 und wurde seitdem kontinuierlich weiterentwickelt. Es existieren zahlreiche Standards mit Datenübertragungsraten von 10 Mbit/s bis zu 40.000 Mbit/s (Stand 2020). Die Übertragung erfolgt stets kabelgebunden und zählt zu den passiven Anschlussarten, d.h. die verwendeten Netzwerkgeräte sind nur aktiv, wenn auch Daten gesendet werden.

Wireless Local Area Network (WLAN) WLAN (IEEE 802.11) ist mit Abstand die bekannteste Technologie für eine drahtlose Datenübertragung. Es existieren zwei unterschiedliche Frequenzbereiche für WLAN, bei dem die Netzwerkeinheiten entweder mit 2.4 GHz oder mit 5 GHz funken. Je

nach Funkstandard sind verschiedene Datenübertragungsraten möglich (Tabelle 3.2):

Standard	Brutto-Übertragungsrate	Netto-Übertragungsrate	Frequenzen [GHz]
802.11	2 Mbit/s	1 Mbit/s	2.4
802.11a	54 Mbit/s	20 – 22 Mbit/s	5
802.11b	11 Mbit/s	5 – 6 Mbit/s	2.4
802.11g	54 Mbit/s	20 – 22 Mbit/s	2.4
802.11h	54 Mbit/s	20 – 22 Mbit/s	5
802.11n	600 Mbit/s	200 – 250 Mbit/s	2.4, 5
802.11ac	1733 Mbit/s	800 – 850 Mbit/s	5

Tabelle 3.2: Datenübertragungsraten für diverse IEEE-Standards für WLAN, adaptiert aus [Baun(2020)]

Bandbreite Die Bandbreite bzw. die Durchsatzrate ist eine Beschreibung der Leistungsfähigkeit eines Netzwerkes. Sie beschreibt den maximalen Durchsatz in Bit pro Sekunde.

Latenz Die Latenz ist eine weitere Beschreibung der Leistungsfähigkeit eines Netzwerkes. Sie beschreibt die notwendige Zeit für die Übertragung einer Nachricht mit definierter Größe von Sender zu Empfänger. Sie ist die Summe von Ausbreitungsverzögerung, Übertragungsverzögerung und Wartezeit. Die Ausbreitungsverzögerung ist hierbei die Entfernung der Kommunikationsteilnehmer geteilt durch das Produkt aus Lichtgeschwindigkeit und dem Ausbreitungsfaktor (z.B. 0.66 für Glasfaser) des Netzwerkes. Die Übertragungsverzögerung ergibt sich aus der Nachrichtengröße dividiert durch die Bandbreite. Wartezeiten entstehen durch Zwischenspeicherung von teilnehmenden Netzwerkgeräten (z.B. Router).

Netzwerkdurchsatz Im Gegensatz zur angegebenen Bandbreite (siehe 3.2.2), kann der tatsächliche Netzwerkdurchsatz geringer ausfallen.

Paketverlustrate Bei der Datenübertragung über eine Netzwerkverbindung kommt es häufig zu dem Verlust von einzelnen Datenpaketen. Die Paketverlustrate gibt hierbei das Verhältnis zwischen verlorenen und gesendeten Paketen an.

Round-Trip-Time (RTT) Die Round-Trip-Time (RTT), beschreibt die nötige Zeit für die Übertragung eines Frames (siehe 3.2.2) von der Quelle zum Ziel und wieder zurück. Ausschlaggebend für die Beurteilung der Netzwerkqualität ist insbesondere die maximale RTT.

Maximum Transmission Unit (MTU) Die MTU definiert die Menge an Nutzdaten, die zwischen zwei Kommunikationsteilnehmenden innerhalb eines Datenframes genutzt werden dürfen und werden in der Sicherungsschicht (siehe 2 definiert). Tabelle 3.3 zeigt Beispiele für bekannte Netzwerktechnologien.

Technologie	MTU
Ethernet	1500 Bytes
ISDN	576 Bytes
WLAN	2312 Bytes

Tabelle 3.3: MTU für unterschiedliche Netzwerktechnologien, adaptiert aus [Baun(2020)]

3.2.3 Internet Protokoll (IP)

UDP Die Abkürzung *UDP* steht für *User Datagram Protocol* und beschreibt ein verbindungsloses Protokoll innerhalb der Transportschicht. UDP verfügt für die Sicherstellung der Datenübertragung lediglich über eine Fehlersumme. Aus diesem Grund wird UDP in der Regel für Anwendungen eingesetzt,

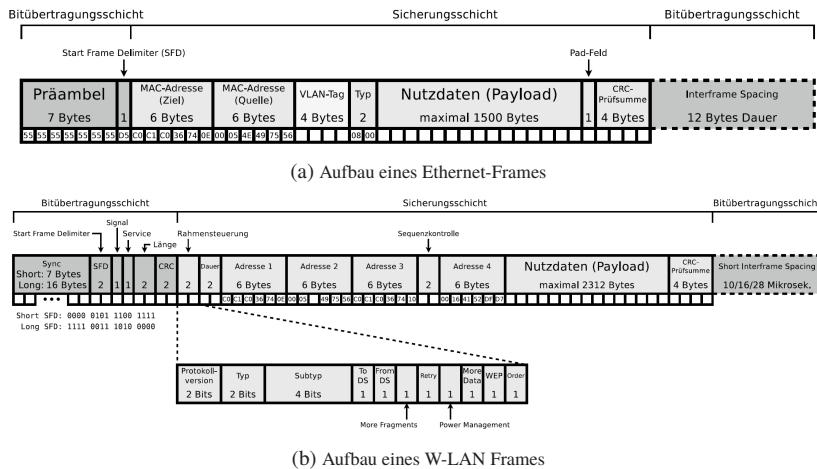


Abbildung 3.4: Unterschied zwischen Ethernet und W-LAN Frame, Quelle: [Baun(2020)]

die nicht sicherheitskritisch sind und bei denen der Verlust einiger gesendeten Frames nicht anwendungsrelevant ist oder diese erneut angefragt werden können. Der UDP-Header ist aus vier 16 Bit Datenfeldern zusammengesetzt (Abbildung 3.5) und hat somit eine Größe von 64 Bit bzw. 8 Byte, die bei jeder Segmentübertragung notwendig sind. Der Pseudo-Header enthält Informationen über die IP-Adresse von Sender und Empfänger sowie Informationen über das verwendete Protokoll und Segmentlänge. Der Pseudo-Header wird bei der Berechnung der Prüfsumme berücksichtigt, allerdings nicht übertragen, um weiteren Overhead zu sparen. Der UDP-Header ist in Abbildung 3.5 dargestellt.

Flusskontrolle Die Flusskontrolle oder Flusssteuerung erlaubt es innerhalb eines Netzwerkes einem Empfänger, der Schwierigkeiten hat die ankommenen Datenframes zu verarbeiten, eine Nachricht an den Sender zu verschicken. Hierbei fordert der Empfänger den Sender auf, die Übertragung für einen Moment zu pausieren und somit einen drohenden Paketverlust durch Überlastung vorzubeugen.

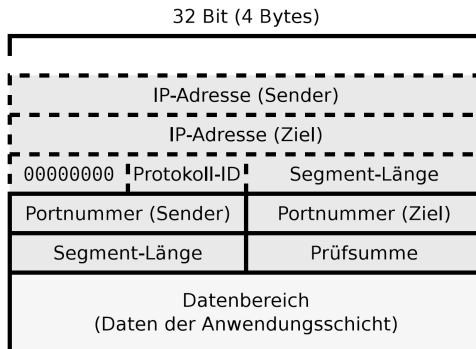


Abbildung 3.5: UDP-Header inklusive Pseudo-Header, [Baun(2020)]

TCP Die Abkürzung *TCP* steht für *Transmission Control Protocol* und ergänzt im Gegensatz zu UDP das IP-Protokoll um Sicherheitsfunktionen bei der Übertragung von Segmenten. TCP garantiert hierbei zum einen die korrekte Reihenfolge und zum anderen die vollständige Übertragung. Mit Hilfe von eindeutigen Sequenznummern können verlorene Segmente erneut gesendet werden, bis deren Ankommen vom Empfänger bestätigt wird. Zudem kommen verschiedene Mechanismen zum Einsatz, die eine Überforderung des Empfängers unterbinden, z.B. Flusskontrolle (siehe 3.2.3). Im Gegensatz zum UDP-Header ist der TCP-Header deutlich komplexer aufgebaut und umfasst eine Größe von 160 Bit bzw. 20 Byte (Abbildung 3.6).

Durch das Feld *Seq-Nummer* wird jedem zu übertragenden Segment eine eindeutige Nummer zugewiesen, die doppelt ankommende Pakete aussortiert und eine korrekte Reihenfolge sicherstellt. Diese wird beim Verbindungsaufbau mit dem Empfänger synchronisiert. Mit dem Feld *Ack-Nummer* wird dem Empfänger eine Referenz zur Empfangsbestätigung mitgeteilt. Der Ablauf für eine Datenübertragung via TCP ist in Abbildung 3.7 dargestellt.

3.3 Datenkompression

Das Ziel der Datenkompression ist die Reduzierung der benötigten Bits zur Speicherung oder Übertragung von Informationen. Die meisten Kompressions-

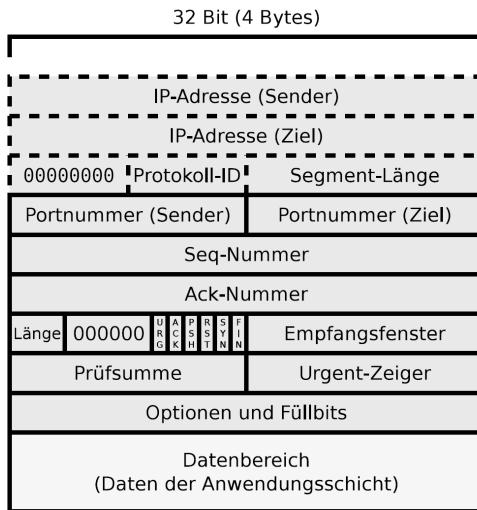


Abbildung 3.6: TCP-Header inklusive Pseudo-Header, [Baun(2020)]

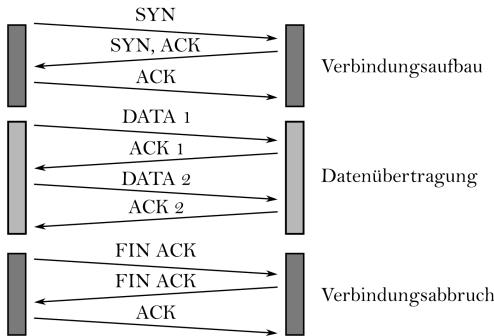


Abbildung 3.7: Schematische Darstellung der Datenübertragung via TCP, adaptiert aus [Baun(2020)]

algorithmen bestehen aus einem *Encoder-Algorithmus*, der die vorliegenden Daten aus dem Ausgangszustand in einen komprimierten Zustand überführt. Dieser komprimierte Zustand ist möglicherweise ohne weiteres schwer bis nicht zu interpretieren. Hierfür wird ein *Decoder-Algorithmus* benötigt, der

die komprimierten Daten teilweise oder vollständig in den Ausgangszustand zurück transformiert. An dieser Stelle sei erwähnt, dass kein Kompressionsalgorithmus existiert, der über verschiedene Anwendungen als „bester“ bezeichnet werden kann. Je nach Anwendungsgebiet und vorliegenden Daten muss ein passender Algorithmus ausgesucht und angewendet werden [Manz(2020)].

Nach Strutz [Strutz(2017)] lässt sich die Bewertung eines Kompressionsalgorithmus durch das *Kompressionsverhältnis* C_R ausdrücken. Quantitativ beschreibt C_R die Relation der Datenmenge des ursprünglichen Signals mit der Datenmenge des codierten Signals:

$$C_R = \frac{\text{Datenmenge(Originalsignal)}}{\text{Datenmenge (nach Kompression)}} \quad (3.20)$$

Grundsätzlich existieren zwei verschiedene Kompressionsverfahren: Die verlustfreie und die verlustbehaftete Kompression. Für verlustbehaftete Verfahren existieren weitere Metriken, die eine Bewertung der Qualität des dekomprimierten Signals beschreiben. Hierzu gehört beispielsweise die *objektive Bewertung* des Betrachtenden über Verzerrung oder Farbtreue.

Weiterhin lässt sich die Qualität entweder durch den mittleren quadratischen Fehler (engl. mean-squared-error, MSE)

$$MSE = \frac{1}{N} \sum_i (x_i - x'_i)^2 \quad (3.21)$$

oder durch das Signal-Rausch-Verhältnis (engl. signal-to-noise ratio, SNR) quantitativ beschreiben. Für die Berechnung des SNR wird die Varianz zwischen Original σ_x^2 und rekonstruiertem Signal σ_e^2 verglichen und in [dB] angegeben:

$$SNR = 10 \cdot \log_{10} \left(\frac{\sigma_x^2}{\sigma_e^2} \right) \quad (3.22)$$

3.3.1 Verlustfreie Kompression

Die verlustfreie Kompression ermöglicht eine vollständige Rekonstruktion der Ausgangsdaten $0 \stackrel{!}{=} MSE \stackrel{!}{=} SNR$. Eine Kompression basiert somit auf einer effizienteren Darstellungsweise im Gegensatz zu den Originaldaten. Die zwei wichtigsten Verfahren sind die *Entropiekodierung* und die *Wörterbuch-Kodierung*.

Entropiekodierung Die bekannteste und am weitesten verbreitetste Entropiekodierung ist die *Huffman-Kodierung* [Huffman(1952)]. Hierbei wird ein Binärbaum von den Blättern zur Wurzel aufgebaut, der einer festen Anzahl an Symbolen in Abhängigkeit ihrer Häufigkeit einen binären Code variabler Länge zuweist. Je häufiger das Vorkommen, desto kürzer der Code.

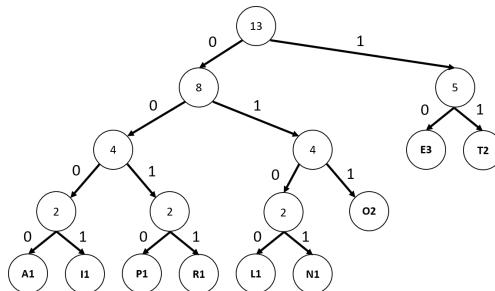


Abbildung 3.8: Beispiel eines Binärbaums für die Huffman-Kodierung für das Wort TELEOPERATION, eigene Darstellung

Abbildung 3.8 zeigt beispielhaft die Kodierung des Wortes TELEOPERATION. Es ergibt sich:

11 10 0100 10 011 0010 10 0011 0000 11 0001 011 0101

Die Kodierung bzw. Dekodierung lässt sich nur durch Verwendung des erstellten Baumes durchführen und ist somit stets abhängig von den vorliegenden Daten, garantiert jedoch eine optimale Entropiekodierung [Strutz(2017)].

Wörterbuch-Kodierung Die Grundidee der Wörterbuch-Kodierung ist das Erstellen eines Wörterbuchs auf Basis der vorliegenden Daten. Anschließend können anstelle von vollständig ausformulierten Symbolen (oder Codes) lediglich Verweise formuliert werden [Manz(2020)]. Die Größe des Wörterbuchs kann bei komplexeren Datenstrukturen schnell ansteigen und eine Suche nach geeigneten Referenzen zu hohen Laufzeiten führen. Aus diesem Grund wird in der Lempel-Ziv-Kodierung (LZ77) [Ziv und Lempel(1977)] ein gleitendes Fenster vorgestellt, das den aktuellen Bereich des Wörterbuchs sowie den aktuellen Bereich der Datei erfasst [Manz(2020)].

Viele gängige Kompressionsverfahren basieren auf einer Kombination aus *Entropiekodierung* und *Wörterbuch-Kodierung*. Hierzu gehören beispielsweise das bekannte ZIP-Archiv oder die PNG-Bilddatenkompression.

3.3.2 Verlustbehaftete Kompression

Im Gegensatz zur verlustfreien Kompression findet bei der verlustbehafteten Kompression entweder eine *Quantisierung* oder eine *Irrelevanzreduktion* statt ($MSE, SNR > 0$). In Anbetracht des dritten Kriteriums, der *objektiven Bewertung*, soll weiterhin ein vergleichbares Ergebnis zu den Ausgangsdaten erreicht werden.

Quantisierung Die Quantisierung überführt ein hochauflösten Eingangssignal in ein niedriger aufgelöstes Ausgangssignal. Die Quantisierung erfolgt dabei meist in vordefinierten Stufen, z.B. die Zeitauflösung eines Sensors oder die Bit-Tiefe eines Datentyps.

Irrelevanzreduktion Ein gutes Beispiel für die Irrelevanzreduktion ist ein Kompressionsschritt des bekannten mp3-Formats [Ruzanski(2006)]. Hierbei wird das Audiosignal im Frequenzbereich betrachtet und für den Menschen nicht hörbare Frequenzen weggescchnitten. Die Kompression ist zwar durch MSE und SNR messbar, allerdings lässt sich durch Hörvergleiche kein Unterschied feststellen.

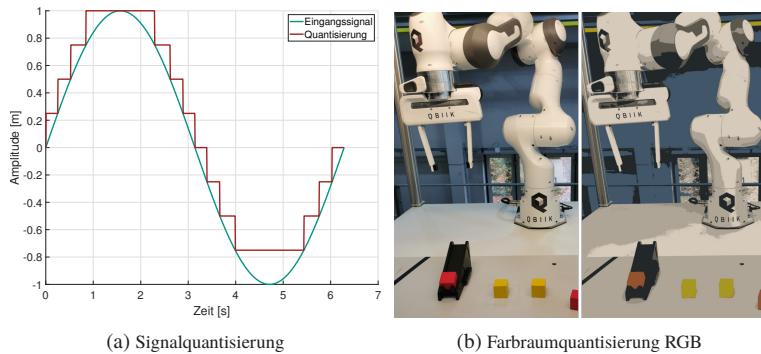


Abbildung 3.9: Beispielhafte Quantisierung eines analogen Signals (links) und des RGB-Farbraums eines Bildes (rechts). In beiden Fällen ist der Effekt der Quantisierung deutlich erkennbar, eigene Darstellung

Chroma-Subsampling Überträgt man die Idee der mp3-Kompression auf die Kompression von Bilddaten, ist eine genauere Betrachtung der menschlichen Farbwarnehmung notwendig. Die Toleranz des menschlichen Auges gegenüber Unschärfe ist für Chrominanz (Farbsignale) deutlich höher als für Luminanz (Helligkeitssignale) [Burger und Burge(2006)]. Findet eine Trennung zwischen Luminanz und Chrominanz statt, lässt sich eine Kompression stärker oder ausschließlich auf die Chrominanz anwenden. Abbildung 3.10 zeigt die einzelnen Komponenten eines im YUV-Farbraum kodiertes Bild.

Die Technologie wurde bereits im Jahr 1950 bei der Einführung des Farbfernsehers verwendet, da die Luminanz bereits dem bestehenden Schwarz-Weiß Signal entsprach und somit eine gute Rückwärtskompatibilität bzw. Erweiterbarkeit bot. Auch heute ist die YUV-Kodierung nach wie vor ein wichtiger Bestandteil bei der Bild- bzw. Videokompression.

Eine Umrechnung aus dem bekannten RGB-Farbraum geschieht nach folgender Vorschrift [Burger und Burge(2006)]:

$$\begin{pmatrix} Y \\ U \\ V \end{pmatrix} = \begin{pmatrix} 0.299 & 0.587 & 0.114 \\ -0.147 & -0.289 & 0.436 \\ 0.615 & -0.515 & -0.100 \end{pmatrix} \cdot \begin{pmatrix} R \\ G \\ B \end{pmatrix} \quad (3.23)$$

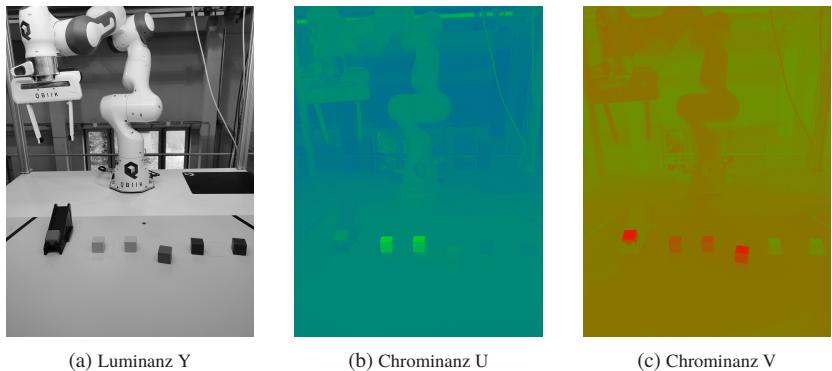


Abbildung 3.10: Darstellung der YUV-Komponenten eines Farbbildes, eigene Darstellung

bzw. umgekehrt über die Inverse aus Gleichung 3.23:

$$\begin{pmatrix} R \\ G \\ B \end{pmatrix} = \begin{pmatrix} 1.000 & 0.000 & 1.140 \\ 1.000 & -0.395 & -0.581 \\ 1.000 & -2.032 & 0.000 \end{pmatrix} \cdot \begin{pmatrix} Y \\ U \\ V \end{pmatrix} \quad (3.24)$$

Die Art der Kompression der Chrominanz wird durch die Parameter J,a,b angegeben und üblicherweise mit $YUV-J:a:b$ notiert.

YUV	4:1:1	4:2:0	4:2:2	4:4:4	4:4:0
Y	=	=	=	=	=
U	1 2 3 4 1 1 1 1	1 2 3 4 1 2 1 2	1 2 3 4 1 2 1 2	1 2 3 4 1 2 1 2	1 2 3 4 1 2 3 4 1 2 3 4
V	a = 1 b = 1	a = 2 b = 0	a = 2 b = 2	a = 4 b = 4	a = 4 b = 0
Aufl. horz.	$\frac{1}{4}$	$\frac{1}{2}$	$\frac{1}{2}$	1	$\frac{1}{2}$
Aufl. vert.	1	$\frac{1}{2}$	1	1	$\frac{1}{2}$

Abbildung 3.11: YUV-Pixelformate, adaptiert aus [Wikimedia(2021)]

J beschreibt den horizontalen Abtastbezug und wird normalerweise als 4 angegeben. Die Anzahl der Chrominanz-Abtastwerte (U, V) wird durch a mit Bezug zur ersten Reihe von J Pixeln beschrieben. Zwischen der ersten und zweiten Reihe von J Pixeln beschreibt b die Anzahl der Änderungen der Chrominanz-Abtastwerte (U,V). Abbildung 3.11 gibt eine Übersicht über die gängigsten YUV-Pixelformate.

Das in dieser Arbeit verwendete NV12-Format (Kapitel 5.2.3 und 5.2.5) basiert auf der YUV-Kodierung 4:2:0. Das bedeutet, dass horizontal und vertikal die Auflösung der Chrominanz im Vergleich zur Luminanz halbiert wird.

3.3.3 Kompression von Videodaten

Bisher wurde lediglich die Kompression von Einzelbildern betrachtet. Für die Kompression von Videodaten, also eine Sequenz aus Einzelbildern, kann neben der Ausnutzung von örtlichen Korrelationen und Irrelevanzen zusätzlich auf eine zeitliche Korrelation zurückgegriffen werden.

Die grundsätzliche Idee der Kompression durch Ausnutzung der zeitlichen Korrelation ist die Wiederverwendung von bereits gesehenen Strukturen, Objekten oder Hintergründen. Bilder, die innerhalb einer Sequenz zur Kompression durch Ausnutzung der zeitlichen Korrelation Verwendung finden, werden als *GOP*, (engl. Group-of-Pictures) bezeichnet. Eine GOP kann aus drei verschiedenen Bildarten (Frames) bestehen: I-Frame, B-Frame und P-Frame [Strutz(2017)].

Der I-Frame bzw. Intraframe verwendet keine zeitliche Korrelationen und stellt somit eine Art Einzelbild dar. Lediglich örtliche Korrelationen innerhalb des Bildes werden für die Kompression verwendet.

Der P-Frame bzw. Prädiktionsframe verwendet als Grundlage entweder einen vorhergehenden P-Frame oder einen I-Frame. Die Vorhersage wird auf Basis berechneter Bewegungsvektoren getroffen und lediglich eine nicht vorhersagbare Veränderung muss übertragen werden. Eine Kompression von örtlichen Korrelationen wird weiterhin durchgeführt.

Der B-Frame bzw. Bidirektionale Prädiktionsframe verwendet als Grundlage sowohl vorhergehende P- bzw. I-Frames als auch nachfolgende. Das Finden von

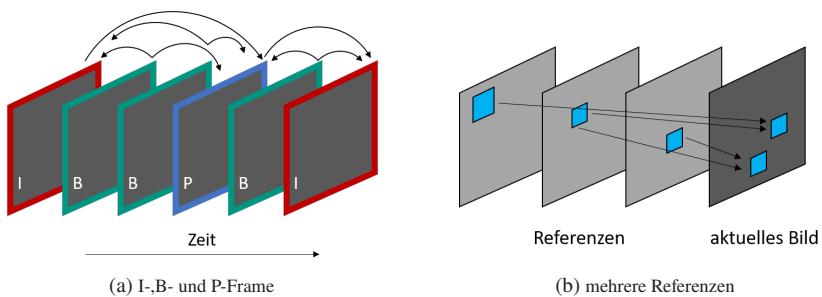


Abbildung 3.12: Ausnutzung zeitlicher Korrelationen (a) durch I-Frame (rot), B-Frame (grün) und P-Frame (blau). Die Verwendung von mehreren Referenzen (b) ist in modernen Videokompressionsalgorithmen möglich, eigene Darstellung

geeigneten Referenzen benötigt zwar entsprechend mehr Rechenleistung, führt aber meist zu besseren Kompressionsraten. Die Verwendung von B-Frames ist für eine Liveübertragung nicht relevant, da die Zusammensetzung von vorhergehenden und nachfolgenden Frames zwangsläufig zu einer Erhöhung der Latenz führt.

Moderne Videokompressionsalgorithmen wie bspw. H264 oder H265 (HEVC) erlauben eine Prädiktion durch Verwendung mehrerer Referenzen (Abbildung 3.12b), die eine Kompression weiter verbessern. Eine genaue Beschreibung der verwendeten Techniken, wie z.B. der diskreten Kosinustransformation für eine anschließende Quantisierung, die Umsetzung des Block-Matchings für die Bewegungsprädiktion etc. findet sich in [Strutz(2017)].

3.4 Anomalie-Erkennung

Das Wort *Anomalie* kommt aus dem Griechischen und bedeutet *Abweichung vom Allgemeinen* oder *Unregelmäßigkeit*. Entsprechend behandelt die Anomalieerkennung das Finden von Mustern, Ausreißern, Besonderheiten oder schlicht unerwarteten Werten innerhalb eines Datensatzes oder in Echtzeitdaten.

Es existiert eine Vielzahl unterschiedlicher Ansätze [Chandola et al.(2009)Chandola, Banerjee, und Kumar], [Mehrotra et al.(2017)Mehrotra, Mohan, und Huang]

die entweder im Falle von a-priori bekannten Anomalien in die Klasse des *Überwachte Lernens* oder bei a-priori unbekannten Anomalien zu der Klasse des *Unüberwachten Lernens* (siehe 3.6) gehören. Die Anomalieerkennung lässt sich in drei Kategorien aufteilen, die im Folgenden erläutert werden sollen:

- Distanzbasierte / Regelbasierte Anomalieerkennung
- Clusteringbasierte Anomalieerkennung
- Modellbasierte Anomalieerkennung

Distanzbasierte/Regelbasierte Anomalieerkennung Die distanzbasierte oder auch regelbasierte Anomalieerkennung gehört zu den einfachsten Arten der Anomalieerkennung. Im Falle von bekannten Grenzwerten können gewisse Distanzen als abnormal bezeichnet und somit eine Regel definiert werden, die während der Laufzeit eines Systems nicht verletzt werden darf. Die Unterscheidung, ob ein bestimmter Messwert p als Anomalie eingestuft wird, kann entweder absolut oder relativ zu weiteren Messpunkten bestimmt werden.

Die Auswertung eines Messpunktes p kann nach [Mehrotra et al.(2017)Mehrotra, Mohan, und Huang] mit einer beliebigen Auswertefunktion α und dem absoluten Schwellwert θ bestimmt werden:

$$\alpha(p) > \theta \quad (3.25)$$

Soll entsprechend überprüft werden, ob die Distanz eines Messpunktes p_i gegenüber weiteren Messpunkten $q_{k..n}$ aus dem Datensatz \mathbb{D} über einem gewissen Schwellwert θ liegt, so kann entweder die Distanz zu allen Punkten als Metrik verwendet werden:

$$\alpha(p) = \sum_{q \in \mathbb{D}} d(p, q) \quad (3.26)$$

oder die Distanz zu den jeweils nächsten Nachbarn ausgewertet werden:

$$\alpha(p) = \min_{q \in \mathbb{D}, q \neq p} d(p, q) \quad (3.27)$$

Am Beispiel eines Kommissionierroboters können das Greifpositionen außerhalb des Arbeitsbereich des Roboters oder zu hohe gemessene Kraftwerte an den Gelenken des Knickarmroboters sein.



Abbildung 3.13: Laserscanner überwacht den Greifprozess des IFL PiRo Roboters [Colling et al.(2017)Colling, Dziedzitz, Hopfgarten, und Markert et al.]

Eine Überwachung des Greifprozesses durch Sensorik ist ebenfalls möglich und kann als Anomalieerkennung fungieren: Die Daten des Laserscanners (Abbildung 3.13) geben hierbei Auskunft, ob ein Greifvorgang oder ein Ablegevorgang erfolgreich verlaufen ist. Darüber hinaus kann ein möglicher Verlust des Objekts während des Greifvorgangs lokalisiert und ggf. korrigiert werden.

Clusteringbasierte Anomalieerkennung Im Gegensatz zur distanzbasierten Anomalieerkennung werden bei der clusteringbasierten Anomalieerkennung die aufgenommenen Datenpunkte nicht unmittelbar miteinander verglichen, sondern zunächst in verschiedene Klassen aufgeteilt. Hierfür eignen sich diverse Clusteralgorithmen wie z.B. *k-Nearest Neighbor* [Ertoz et al.(2002)Ertoz, Steinbach, und Kumar] oder *k-Means Clustering* [Lloyd(1982)]. Anschließend können die bei der distanzbasierten Anomalieerkennung vorgestellten Metriken verwendet werden, um entweder Anomalien innerhalb einer Klasse zu finden oder die Distanz zwischen dem Schwerpunkt der generierten Klassen zu untersuchen.

Für das Beispiel des Kommissionierroboters könnten somit beispielsweise mehrere Ablegepositionen innerhalb des Arbeitsraums gruppiert und auf ihre Gültigkeit überprüft werden.

Modellbasierte Anomalieerkennung Die modellbasierte Anomalieerkennung grenzt sich zu den beiden bereits vorgestellten Methoden ab, da hierbei nicht nur einzelne Datenpunkte und deren Werte betrachtet werden, sondern vielmehr eine Art unterliegendes Prozessverständnis in Form eines Modells hinterlegt wird. Das Modell ist also in der Lage, die Entstehung der Messpunkte - zumindest bis zu einem gewissen Grad - nachzuvollziehen. Passen nun einzelne Werte nicht zu dem vorliegenden Modell, kann von einer Anomalie ausgegangen werden. Die Struktur des Modells kann hierbei entweder zur Untersuchung von einzelnen Messpunkten oder für ganze Zeitreihen von Messungen ausgelegt werden. Des Weiteren erlauben die Modelle eine Bewertung von Daten, die sich nicht auf herkömmlichen Wege separieren lassen. Als Beispiel seien lineare Regressionsmodelle, Hidden-Markov-Modelle (HMMs) sowie nicht-lineare Regressionsmodelle - zu denen auch die Künstlichen neuronalen Netze (siehe 3.6 gehören, genannt.

Um das Beispiel des Kommissionierroboters erneut aufzugreifen, können durch das unterliegende Prozessverständnis der Modellbasierten Anomalieerkennung deutlich komplexere Zusammenhänge des Greifprozesses untersucht werden. Eine Betrachtung von vielen Sensordaten, wie beispielsweise einer Kamera in Kombination zu den Kraftsensoren des Roboters, wird hiermit möglich. Bereits gesehene Objekte oder durchgeführte Bewegungen können abgespeichert und Abweichungen als Anomalie erkannt werden.

3.5 Kameramodell

Die in einem Kamerabild abgebildeten Pixel entstehen durch eine Projektion der Echtwelt durch eine Kameralinse auf einem Kamerasensor. Das Lochkameramodell beschreibt diesen Zusammenhang mit *extrinsischen* und *intrinsischen* Kameraparametern.

Insgesamt werden nach [Jain et al.(1995)Jain, Kasturi, und Schunck] bei einer Kamera vier unterschiedliche Koordinatensysteme betrachtet (Abbildung 3.14). Das *Weltkoordinatensystem* (χ, η, ζ) beinhaltet die Position jeglicher Objekte sowie die Position der Kamera in Relation zu ihnen. Der Koordinatenursprung kann hierbei frei gewählt werden. Die Lochblende stellt den Ursprung eines weiteren Koordinatensystems dar, dem *Kamerakoordinatensystem* (x, y, z). Dieses ist abhängig von der Position der Kamera in Weltko-

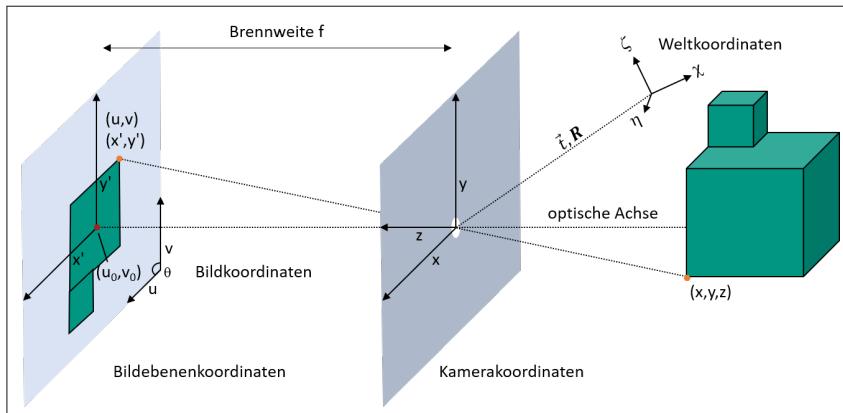


Abbildung 3.14: Zusammenhang von Weltkoordinaten und Bildkoordinaten, eigene Darstellung

ordinaten sowie von der Blickrichtung der Kamera. Die beiden letzten Koordinatensysteme, das *Bildebenenkoordinatensystem* (x', y') und das *Bildkoordinatensystem* (u, v) sind zweidimensional und stellen die Projektion der dreidimensionalen Punkte des Kamerakoordinatensystems auf der Sensorebene der Kamera (Bildebenenkoordinatensystem) sowie die tatsächlich gemessenen Pixel eines Bildes (Bildkoordinatensystem) dar.

Unter Verwendung von homogenen Koordinaten kann eine Transformation von Weltkoordinaten zu Kamerakoordinaten durch eine Translation mit dem Translationsvektor \vec{t} und eine Rotation mit einer Rotationsmatrix \mathbf{R} in Matrixschreibweise realisiert werden:

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \mathbf{R} \begin{pmatrix} \chi \\ \eta \\ \zeta \end{pmatrix} + \vec{t} = (\mathbf{R} | \vec{t}) \begin{pmatrix} \chi \\ \eta \\ \zeta \\ 1 \end{pmatrix} \quad (3.28)$$

Die Rotationsmatrix \mathbf{R} und der Translationsvektor \vec{t} werden auch als *extrinsische Kameraparameter* bezeichnet. Die Projektion des Kamerakoordinatensys-

tem auf das Bildebenenkoordinatensystem kann mit Hilfe des Strahlensatzes und der Brennweite f der Kamera hergeleitet werden:

$$\frac{x}{z} = \frac{x'}{f} \quad (3.29)$$

$$\frac{y}{z} = \frac{y'}{f} \quad (3.30)$$

In Matrixschreibweise können Gleichung 3.29 und 3.30 zusammengefasst werden:

$$z \begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} f & 0 \\ 0 & f \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \quad (3.31)$$

Für die Folgetransformation von Bildebenenkoordinatensystem zu Bildkoordinatensystem sind fünf weitere Parameter notwendig, die als *intrinsische Parameter* bezeichnet werden: die horizontale Auflösung der Kamera α , die vertikale Auflösung der Kamera β , der Scherwinkel zwischen den Koordinatenachsen (u, v) des Bildkoordinatensystems θ sowie die Koordinaten des Hauptpunktes (u_0, v_0) .

Werden die *intrinsischen Kameraparameter* in der Kameramatrix \mathbf{K} zusammengefasst, kann eine Transformationsvorschrift von Weltkoordinaten in Bildkoordinaten aufgestellt werden:

$$z \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \underbrace{\begin{pmatrix} f\alpha & -f\beta \cot \theta' & u_0 \\ 0 & \frac{f\beta}{\sin \theta'} & v_0 \\ 0 & 0 & 1 \end{pmatrix}}_{:= \mathbf{K}} \mathbf{R} \begin{pmatrix} \chi \\ \eta \\ \zeta \end{pmatrix} + \vec{t} \quad (3.32)$$

3.6 Maschinelles Lernen

Maschinelles Lernen ist ein Oberbegriff für Algorithmen, die anhand von Beispieldaten oder Erfahrungswerten aus der Vergangenheit trainiert bzw. optimiert werden [Alpaydin(2019)]. Hierzu gehören, inzwischen als klassische Algorithmen des maschinellen Lernens bezeichnet, die Regressionsanalyse, Support-Vektor-Maschinen (SVM), Entscheidungsbäume sowie diverse Algorithmen zur Klassifizierung von Daten. Durch Fortschritte im Bereich von effizienteren Trainingsmethoden, der parallelen Datenverarbeitung sowie Geschwindigkeitsverbesserungen von CPUs und GPUs haben Künstliche Neuronale Netze (KNNs) den Bereich des maschinellen Lernens beinahe vollständig eingenommen. Sie liefern in nahezu allen Bereichen bessere Ergebnisse und bieten auf Basis ihrer Netzwerkstruktur eine einfache Möglichkeit der Skalierung von gegebenen Aufgabenbereichen und stellen somit den fundamentalen Bestandteil der vorliegenden Arbeit für kontinuierlich lernende Systeme dar.

Künstliche Neuronale Netze Künstliche Neuronale Netze (KNN) oder umgangssprachlich neuronale Netze (NN) bezeichnen eine Klasse von Algorithmen, die sich an der menschlichen Gehirnstruktur, also biologischen neuronalen Netzen, orientieren. Hierbei existieren nach [Alpaydin(2019)] eine Vielzahl von einzelnen Neuronen, die durch eine Art Netzwerk miteinander verbunden sind. Diese kommunizieren untereinander durch Signale. Im menschlichen Gehirn geschieht dies durch Synapsen, die elektrische oder chemische Signale zwischen einzelnen Neuronen transportieren. Bei den Künstlichen Neuronalen Netzen (KNN) wird dieses Verhalten durch eine Verkettung numerischer Werte mit Gewichten, einer Übertragungsfunktion, einer Aktivierungsfunktion (Abbildung 3.16) sowie einer spezifischen Gewichtung der Neuronen realisiert.

Perzeptron Das grundlegende Verarbeitungselement stellt das Perzeptron dar (Abbildung 3.15).

Die Ausgabe y stellt in diesem Fall die gewichtete Summe aller Eingaben x_j , $j = 1, \dots, d$ multipliziert mit den jeweiligen Gewichten w_j , $j = 1, \dots, d$ dar.

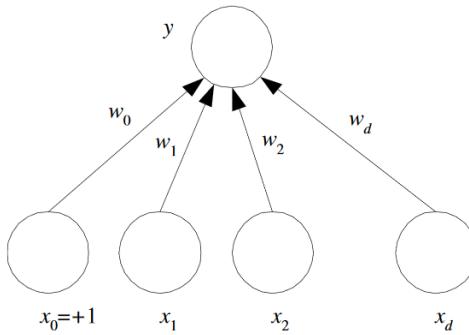


Abbildung 3.15: Das Perzeptron als grundlegendes Verarbeitungselement von Künstlichen Neuronalen Netzen (KNN). Die Eingänge $x_j, j = 1, \dots, d$ werden über die Gewichte w_j zur Ausgabe y verbunden. [Alpaydin(2019)]

$$y = \sum_{j=1}^d w_j x_j + w_0 \quad (3.33)$$

wobei w_0 als Ordinate bezeichnet wird und durch $x_0 = 1 \cdot w_0$ bestimmt wird. Durch die Verwendung der Vektoren $\mathbf{w} = [w_0, w_1, \dots, w_d]^T$ und $\mathbf{x} = [1, x_1, \dots, x_d]^T$ lässt sich Gleichung 3.33 als Skalarprodukt beschreiben:

$$y = \mathbf{w}^T \mathbf{x} \quad (3.34)$$

Mit Hilfe eines einzelnen Perzeptrons mit lediglich einem Eingang $x_1 = x$ erhält man die allgemein bekannte Geradengleichung

$$y = w_1 \cdot x + w_0 \quad (3.35)$$

und ist somit in der Lage, eine lineare Funktion auf Basis einzelner Stichproben zu klassifizieren. Besitzt das Perzeptron weitere Eingänge entsteht eine (Hyper-)Ebene und erlaubt eine Klassifizierung der Eingangsdaten in zwei unterschiedliche Klassen - sofern diese linear trennbar sind [Alpaydin(2019)]. An dieser Stelle sei erwähnt, dass das ursprüngliche Perzeptron-Modell eine bi-

binäre Schwellwert-Funktion als Aktivierungsfunktion φ besitzt [Haykin(2009)] und somit lediglich eine binäre Klassifizierung der Eingangsdaten erlaubt (Abbildung 3.16).

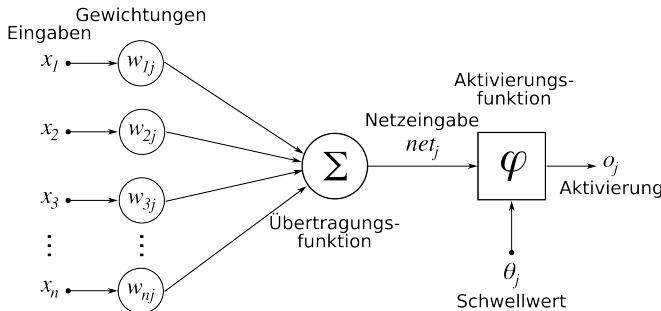


Abbildung 3.16: Neuronen mit Übertragungsfunktion und Aktivierungsfunktion φ [Wikimedia(2020b)]

Mehrschicht-Perzeptron-Netzwerk Verkettet man eine Vielzahl an Perzeptronen (Abbildung 3.17) und fasst diese in einer *Eingangsschicht*, einer oder mehreren *versteckte(n) Schicht(en)* sowie einer *Ausgangsschicht* zusammen, erhält man ein sogenanntes Mehrschicht-Perzeptron-Netzwerk (engl. Multi-Layer-Perceptron, MLP). Im Gegensatz zu einem einzelnen Perzeptron ist ein MLP-Netzwerk in der Lage, deutlich komplexere Klassifizierungen der Eingangsdaten vorzunehmen. Allerdings ist es zwingend erforderlich, die Aktivierungsfunktion φ nicht als binäre- sondern als lineare Funktion zwischen 0 und 1 zu verwenden. Hierdurch wird eine lineare Approximation an die vorliegenden Daten erreicht [Ertel(2013)]. Man spricht somit auch von linearen Neuronen. Mit einer hohen Anzahl an versteckten Schichten werden die Netzwerke als *Tiefe neuronale Netze* (engl. *Deep Neural Networks*) bezeichnet.

Neben der linearen Aktivierungsfunktion φ existieren viele weitere Aktivierungsfunktionen, die das Lernverhalten von Neuronalen Netzen stark beeinflussen (Abbildung 3.18).

Eine Vielzahl an Untersuchungen über den Einfluss der Aktivierungsfunktion auf die Leistungsfähigkeit von Künstlichen Neuronalen Netzen lassen sich in der Literatur finden [Mhaskar und Micchelli(1993)], [Ittiyavirah

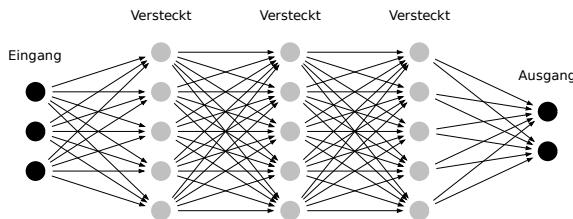


Abbildung 3.17: Schematische Darstellung eines Mehrschicht-Perzeptron-Netzwerks mit einer Eingangsschicht, mehreren versteckten Schichten und einer Ausgangsschicht, eigene Darstellung

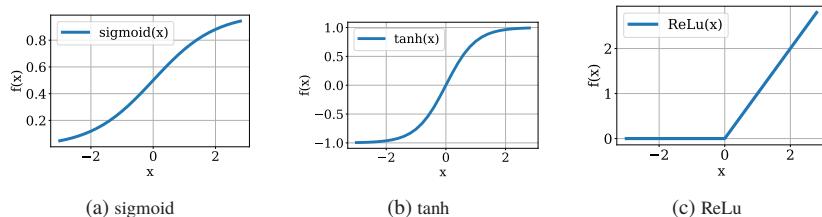


Abbildung 3.18: Vergleich der am häufigsten verwendeten Aktivierungsfunktionen, eigene Darstellung

et al.(2013)Ittiyavirah, Jones, und Siddarth], [Sharma et al.(2020)Sharma, Sharma, und Athaiya]. Es konnte gezeigt werden, dass bei kleineren Netzwerken sowie Netzwerken für Klassifizierungen eine Sigmoid-Funktion zu präferieren ist, während die ReLu-Funktion im Allgemeinen eine bessere Leistungsfähigkeit zeigt. Nachteil der Sigmoid- und Tanh-Funktion ist ein Gradient nahe Null für hohe Werte, der das Lernen deutlich verlangsamt. Es lässt sich jedoch (zumindest bisher) keine Metrik bei Tiefen Neuronalen Netzen für die Auswahl der Aktivierungsfunktion beschreiben. Die Leistungsfähigkeit ist somit stets von Aufgabe, Netzwerkstruktur und Trainingsdaten abhängig [Sharma et al.(2020)Sharma, Sharma, und Athaiya].

Training von Künstlichen Neuronalen Netzen Ein erfolgreicher Lernprozess kann als Optimierung einer gewissen Fehlerfunktion $f(e)$ verstanden werden, die durch die Eingangsdaten i zu den Ausgangsdaten o führt und mit den Zieldaten t verglichen wird. Das KNN kann hierbei als eine Art unbekannte Funktion $f(i)$ angesehen werden:

$$f(i) = o \quad (3.36)$$

$$o \stackrel{!}{=} t \quad (3.37)$$

$$e = f(t, o) \quad (3.38)$$

Je nachdem ob die Zieldaten bekannt sind, werden grundsätzlich zwei verschiedene Trainingsarten unterschieden [Berry et al.(2020)Berry, Mohamed, und Yap]:

- Überwachtes Lernen (engl. supervised learning)
- Unüberwachtes Lernen (engl. unsupervised learning)

Beim *überwachten Lernen* ist die gewünschte Ausgabe des KNNs für jeden Eingang bekannt. Für eine Klassifizierung muss also für jedes Element die zugehörige Klasse der Eingangsdaten im Vorhinein bekannt sein. Das Wissen über den gewünschten Ausgang wird auch als *label* bezeichnet, für einen ganzen Datensatz ist somit von einem *gelabelten Datensatz* die Rede.

Im Gegensatz zum *überwachten Lernen* sind beim *unüberwachten Lernen* die Zieldaten unbekannt. Als Beispiel sei die Klassifizierung der Eingangsdaten zu einer bisher unbekannten Anzahl an Klassen genannt. Die Zuordnung wird vom Netzwerk selbst bestimmt und gelernt.

Die Anpassung der Gewichte des KNNs, also der tatsächliche Lernschritt, geschieht meist¹ durch *Backpropagation* [Rumelhart et al.(1986)Rumelhart,

¹ Die Suche nach einer besseren Alternative, insbesondere durch die Problematik von Gradienten nahe Null bei hohen Werten sowie keinen Beweis für die Ähnlichkeit zu biologischen Neuronalen Netzen, ist Bestandteil aktueller Forschung [Bengio et al.(2016)Bengio, Lee, Bornschein, Mesnard, und Lin], [Ma et al.(2019)Ma, Lewis, und Kleijn], [Choromanska et al.(2019)Choromanska, Cowen, Kumaravel, und Luss et al.]

Hinton, und Williams]. Der bereits im Jahr 1986 vorgestellte Algorithmus basiert auf der Optimierung der Gewichte auf Basis des Gradientenabstiegs im Gewichtsraum des KNNs. Ausgehend von dem Fehler zwischen Ausgangsdaten des Netzwerkes und den gewünschten Zieldaten wird der Einfluss einzelner Neuronen durch deren Ableitung untersucht. Neuronen mit großen Einfluss werden weiter verstärkt, Neuronen mit niedrigem Einfluss weiter geschwächt. Schicht für Schicht wird der Fehler also rückwärts propagiert (engl. back propogated) und die Gewichte entsprechend angepasst.

4 Konzept für Teleoperation und kontinuierliches Lernen

Das folgende Kapitel 4 stellt einen Ablaufplan vor, wie automatisierte Systeme durch den Faktor Mensch ergänzt werden können und beantwortet damit die erste forschungsleitende Fragestellung:

Wie kann die menschliche Überlegenheit bei der Dateninterpretation in komplexe Prozessabläufe von cyber-physischen Systemen integriert werden?

Die vorgestellten Ausführungen wurden während der Arbeit mit einem stationären Kommissionierroboter entwickelt, eine Übertragung der Ansätze auf weitere CPS, wie bspw. ein Fahrerloses Transportfahrzeug (FTF), ist durchaus denkbar.

Weiterhin wird erstmals eine Kategorisierung der menschlichen Assistenz durch Teleoperation beschrieben und die zweite forschungsleitende Fragestellung beantwortet:

Fragestellung 2: In welche Stufen lässt sich Teleoperation für cyber-physische Systeme kategorisieren?

4.1 Vorstellung des erarbeiteten Konzepts

Der Grundgedanke für das Miteinbeziehen menschlicher Fähigkeiten bei der Dateninterpretation eines bereits automatisierten cyber-physischen Systems ist das Aufrechterhalten der Funktion bei unbekannten Systemzuständen. Das erarbeitete Konzept besteht aus insgesamt fünf Modulen, die im Falle einer drohenden Systemstörung einen menschlichen Eingriff aus der Ferne ermöglichen und zudem ein Framework für das kontinuierliche Lernen bilden. Die

Module sind im folgenden aufgelistet und deren Verbindungen in Abbildung 4.1 dargestellt.

1. Automatisierte Funktion des CPS
2. Anomalieerkennung
3. Teleoperation
4. Lerndatengenerierung
5. Lernen

Ausgehend von der *automatisierten Funktion des CPS* wird eine *Anomalieerkennung* als überwachende Instanz hinzugefügt. Diese beobachtet die Interaktion des Systems mit der Umgebung und ist in der Lage zu unterscheiden, ob eine Aufgabe erfolgreich oder nicht erfolgreich durchgeführt wurde oder durchgeführt werden kann. Im Falle einer positiven Bewertung der *Anomalieerkennung* kann das System zugetragene Aufgaben automatisiert durchführen.

In Folge einer negativen Bewertung der *Anomalieerkennung* kann durch das Modul *Teleoperation* ein Mensch um Hilfe gebeten werden. Dieser wird nun temporär befähigt, dem System auf unterschiedlichen Ebenen (siehe Kapitel 4.7) zu assistieren. Nachdem der Teleoperator die Aufgabe für das System gelöst hat, kann dieses mit der automatisierten Funktion fortfahren.

Das Modul *Lerndaten Generierung* zeichnet hierbei diverse Eingangsdaten der *Automatisierten Funktion* und der *Teleoperation* auf.

Die aufgezeichneten Daten werden anschließend dem Modul *Lernen* bereitgestellt, das daraufhin die *Automatisierte Funktion* und/oder die *Anomalieerkennung* aktualisiert bzw. verbessert.

Der beschriebene Zusammenhang ist in Abbildung 4.1 dargestellt und die einzelnen Module werden zunächst generisch beschrieben. Eine konkrete Umsetzung der Module für den Einsatz mit einem Knickarmroboter findet sich in den nachfolgenden Kapiteln 5 und 6.

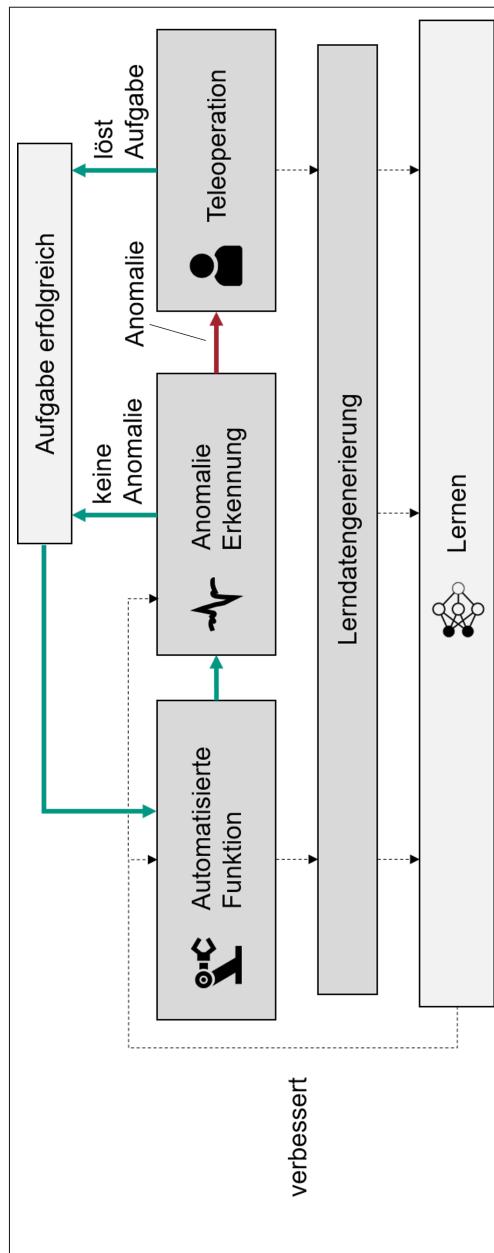


Abbildung 4.1: Konzept für Teleoperation und kontinuierliches Lernen, eigene Darstellung

4.2 Automatisierte Funktion

Aufgabe der automatisierten Funktion ist es, das System auf Basis von Sensorignalen und Algorithmen selbstständig zu regeln. Die eingesetzten Algorithmen müssen, zumindest in Teilen, aus dem Bereich des maschinellen Lernens (siehe Kapitel 3.6) kommen, um eine Funktionsverbesserung auf Basis von Lerndaten zu ermöglichen. Im Allgemeinen basiert die Funktionsweise bereits auf einem initialen Trainingsdatensatz, mit dem die automatisierte Funktion angelernt wurde.

Als Sonderfall seien einzelne Parameter genannt, die zunächst von Hand vorgegeben und entsprechend der gewonnenen Trainingsdaten / Erfahrungswerte iterativ angepasst werden können.

4.3 Anomalieerkennung

Aufgabe der Anomalieerkennung ist es, die automatisierte Funktion des CPS zu überwachen und deren Aktionen zu bewerten. Die Komplexität der Anomalieerkennung kann, je nach Aufgabengebiet und in Abhängigkeit der vorliegenden Daten, gewählt werden. Entsprechend können auch mehrere Anomalieerkennungen (siehe Kapitel 3.4) parallel aktiv sein.

Im Gegensatz zu der in der Literatur üblichen Beschreibung der autonomen Funktion durch unterschiedliche Stufen oder qualitative Handlungskriterien ([Sheridan und Verplank(1978)], [Beer et al.(2014)Beer, Fisk, und Rogers]) erlaubt die Kombination aus *Anomalieerkennung* und *Automatisierter Funktion* einen permanenten Wechsel des Grades der Automatisierung. Die Entscheidung über Erfolg oder Misserfolg einer automatisierten Funktion in dynamischen Umgebungen hängt häufig von kleinen Details ab. In einem Moment ist das System in der Lage, simple Aufgaben wie die Erkennung eines bestimmten Objekts vollständig automatisiert zu erledigen. Allerdings könnte bei veränderten Randbedingungen die Situation eine völlig andere sein: Reflexionen, blendendes Licht, Hindernisse entlang einer Trajektorie etc. sind nur einige Beispiele, die den Grad der Automatisierung eines CPS temporär herabsetzen können. Eine permanente Rückkopplung zwischen *Anomalieerkennung* und *Automatisierter Funktion* ist zwingend erforderlich, um die Fähigkeiten des

CPS richtig einzuschätzen und somit die aktuell geltende Stufe der Automatisierung zu bestimmen.

4.4 Teleoperation

Die *Teleoperation* stellt die Verbindung zwischen Mensch und CPS dar. Hierbei werden die Umgebungsdaten entweder mit Hilfe der verwendeten Sensoren aus der *Automatisierten Funktion* oder durch zusätzlich angebrachte Sensoren aufgezeichnet. Anschließend werden diese an den Teleoperator übertragen. Dieser kann die Daten auf verschiedene Arten visualisieren und sich ein Bild von der vorliegenden Situation machen. Die Befehle zur Lösung der Aufgabe werden nun vom Teleoperator an das CPS zur mittelbaren oder unmittelbaren Ausführung übertragen. Eine detaillierte Beschreibung der Umsetzung der *Teleoperation* erfolgt in Kapitel 5.2.

4.5 Lerndatengenerierung

Die Aufgabe der *Lerndatengenerierung* besteht im Ableiten relevanter Informationen zur weiteren Verwendung bzw. Verbesserung der *automatisierten Funktion*. Zum einen können diese Daten während des automatisierten Betriebs aufgezeichnet werden, um ein bereits funktionierendes System robuster zu machen und gut ausgeführte Aktionen weiter zu fördern. Zum anderen können diese auch aus der *Teleoperation* abgeleitet werden und somit das Expertenwissen des Teleoperators mit jedem Eingriff in die Funktionalität der *automatisierten Funktion* einfließen lassen.

4.6 Lernen

Das eigentliche Lernen, also der Verbesserungsprozess der automatisierten Funktion, findet mit Hilfe der gesammelten Daten der Lerndatengenerierung statt.

Das Lernen kann zu verschiedenen Zeitpunkten stattfinden:

- Nach n ausgeführten Aktionen
- beim unterschreiten einer definierten Erfolgsquote der automatisierten Funktion
- manuell ausgeführt durch einen (Tele-)Operator

Eine unmittelbare Verbesserung der automatisierten Funktion lässt sich durch direkte Nachahmung der vorgemachten Aufgabe bzw. direkte (Parameter-) Anpassung der steuernden Algorithmen realisieren.

4.7 Verbindung von Mensch und Maschine

Die vorgestellten Module und deren Verknüpfung erlauben es einem Menschen, in die Prozessabläufe eines CPS einzugreifen, sollte dieses Hilfe benötigen. In welchem Umfang eine Hilfestellung notwendig ist, hängt vom Einsatzgebiet und vom vorliegenden Problem des CPS ab. Eine Kategorisierung der Hilfestellung bzw. Teleoperation in Abhängigkeit der eingesetzten Peripherie und menschlichen Fähigkeiten ist bisher nicht existent und soll im Folgenden definiert werden.

Das menschliche Verhalten wird durch externe Einflüsse, die durch die *sensorischen Fähigkeiten* wahrgenommen werden, mit dem bereits gelernten *Wissen* durch *kognitive* und *soziale Fähigkeiten* verknüpft. Diese Handlungsableitung wird als *Intelligenz* bezeichnet.

Die Verhaltengenerierung eines modernen CPS ist analog zu der des Menschen: Sensoren liefern aktuelle Informationen über die Umgebung des CPS und durch das Zusammenspiel von den durch die *Sensoren* gewonnen *Daten* und den implementierten *Algorithmen* lässt sich ein gewünschtes Verhalten des CPS berechnen. Ein reaktives Zusammenspiel kann als *künstliche Intelligenz* bezeichnet werden.

Ziel der *Teleoperation* ist es, das (meist) überlegene, menschliche Verhalten bestmöglich an das CPS zu übertragen. Die Übersetzung zwischen menschlichen und maschinellen Fähigkeiten übernimmt die *Peripherie*.

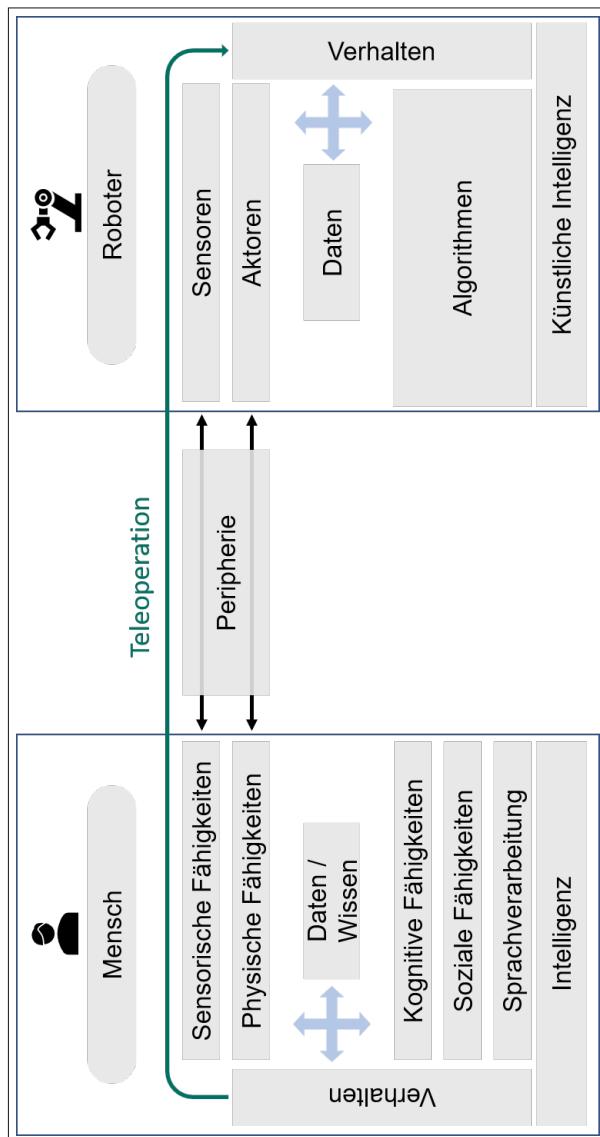


Abbildung 4.2: Verbindung von Mensch und Maschine durch Peripherie, eigene Darstellung

Durch Eingabe-Peripheriegeräte, wie bspw. die Computermaus, einen Touchscreen (sowohl Eingabe- als auch Ausgabe-Peripheriegerät), einen Joystick etc. können Befehle durch physische Interaktion in computerverständliche Sprache übersetzt werden.

Umgekehrt können mit Ausgabe-Peripheriegeräten, wie dem Computermonitor, dem Touchscreen, einer VR-Brille, etc. Signale des Computers in Signale gewandelt werden, die der Mensch mit Hilfe seiner *sensorischen Fähigkeiten* verstehen und interpretieren kann.

Die Verbindung von Mensch und Maschine wird in Abbildung 4.2 mit den, in Kapitel 2.3 bereits vorgestellten, fünf Kategorien verknüpft. Die *sensorischen Fähigkeiten* des Menschen lassen sich durch *Sensoren*, die *Physischen Fähigkeiten* entsprechend durch *Aktoren* an einem CPS abbilden. Während der Mensch weiterhin über *Kognitive Fähigkeiten*, *Soziale Fähigkeiten* und *Sprachverarbeitung* verfügt, lässt sich deren Gegenstück am CPS mit nur einer Kategorie beschreiben: den *Algorithmen*.

Es ist also naheliegend, eine Kategorisierung der Teleoperation basierend auf der eingesetzten Peripherie und somit unmittelbar mit den eingesetzten *sensorischen* und *aktorischen Fähigkeiten* des Menschen zu beschreiben.

Durch Unterscheidung zwischen 2D-Peripherie (Computermonitor, Computermaus) und 3D-Peripherie (AR/VR-Geräte) kann der beschriebene Zusammenhang zwischen menschlichen Fähigkeiten mit *Sensoren* und *Aktoren* des CPS technisch abgegrenzt werden.

Als weiteres Kriterium für die Einstufung der Teleoperation wird die zeitliche Verzögerung der Hilfestellung hinzugezogen. Je nach Aufgabenstellung des Teleoperators kann ein Kamerabild, das einige Minuten oder Stunden in der Vergangenheit liegt für eine angemessene Handlungsempfehlung ausreichen. Ist die Aufgabenstellung zeitkritisch oder eine direkte Steuerung des CPS notwendig, so wird eine Echtzeitübertragung zwischen CPS und Mensch notwendig, um eine angemessene Steuerung des Teleoperators zu gewährleisten.

Abbildung 4.3 visualisiert vier abgeleitete Stufen der Teleoperation in Abhängigkeit der erforderlichen Netzwerkkommunikation und eingesetzten Peripherie:

Stufe	Kommunikation	Peripherie	Autonomie
1	zeitverzögert	2D	mittel-hoch
2	Echtzeit	2D	mittel
3	zeitverzögert	3D	niedrig
4	Echtzeit	3D	keine

Abbildung 4.3: Die Stufen der Teleoperation in Abhängigkeit der eingesetzten Peripherie und Netzwerk-Kommunikation sowie der benötigten Autonomie des CPS, eigene Darstellung

Mit jeder Stufe steigen kontinuierlich die Anforderungen an die Kommunikation und die eingesetzte Sensorik am CPS. Den Stufen der Teleoperation entgegengesetzt verlaufen die Anforderungen an die Autonomie des Systems.

Während bei einer Teleoperation mit 2D Eingabe- und Ausgabe-Peripherie (Stufe 1 und Stufe 2) der Teleoperator in seiner Handlungsfähigkeit beschränkt ist, muss das System die fehlenden Informationen selbst herleiten. Eine direkte Steuerung des CPS ist somit für die Stufen 1 und 2 ausgeschlossen. Bei der Teleoperation mit 3D Eingabe- und Ausgabe-Peripherie hingegen können, aufgrund des erhöhten Handlungsspielraums, weitere Aufgaben an den Teleoperator abgegeben werden (siehe Kapitel 2.4).

Entsprechend steigen mit jeder Stufe die Anforderungen an die *sensorischen Fähigkeiten* und *physischen Fähigkeiten* des teleoperierenden Menschen und stellen somit eine Art Aufwand für die Hilfestellung dar.

Im weiteren Verlauf der Arbeit werden häufig die Stufen 1/2 und 3/4 auf Grund ihrer technischen Ähnlichkeit in Bezug zur eingesetzten Peripherie gemeinsam betrachtet. Die gemeinsame Erwähnung soll keinesfalls die weitere Unterteilung in Bezug zur eingesetzten Netzwerkkommunikation in Frage stellen.

5 Umsetzung der Teleoperation

Im folgenden Kapitel wird eine konkrete Umsetzung der notwendigen technischen Module der Teleoperation sowie deren Implementierung vorgestellt. Das Kapitel beschreibt somit das Modul *Teleoperation* des bereits vorgestellten Konzepts eines kontinuierlich lernenden CPS aus Kapitel 4.

5.1 Vorstellung des entwickelten Demonstrators

Im Rahmen des Forschungsprojekts QBIIK [BMW(2020a),5] und dieser Dissertation wurde ein speziell zur Erforschung der Teleoperation konzipierter Demonstrator am Institut für Fördertechnik und Logistiksysteme (IFL) des Karlsruher Instituts für Technologie (KIT) in Eigenleistung entwickelt und gebaut. Eine Übersicht über die verbauten Komponenten ist in Abbildung 5.1 dargestellt.

Der verbaute Knickarmroboter *Franka emika panda* gehört zur Gruppe der kolaborativen Roboter und besitzt sieben Freiheitsgrade. Der verwendete Greifer ist ebenfalls von der Firma Franka emika und wurde für ein besseres Greifen mit verlängerten Fingern ausgestattet. Zu greifende Teile werden in zwei Kleinladungsträgern (KLT) vom Typ 14/6-2H vor dem Roboter platziert. Eine gleichmäßige Ausleuchtung der Ladungsträger wird durch zwei verbaute LED-Streifen an der Decke des Demonstrators realisiert.

Des Weiteren verfügt der Demonstrator über drei verschiedene Kameras, von denen zwei im Folgenden gemeinsam als *Automationskamera* und eine als *Teleoperationskamera* bezeichnet werden.

Die Automationskamera besteht aus einer Kombination aus Farbkamera und Tiefenbildkamera. Die Farbkamera (IDS UI-5270CP-C-HQ Rev. 2, [IDS(2022b)]) besitzt eine Auflösung von 3.2 Megapixel (2056 x 1542) und ein Objektiv mit 8mm Brennweite. Die Tiefenbildkamera (IDS Ensenso N35-802-

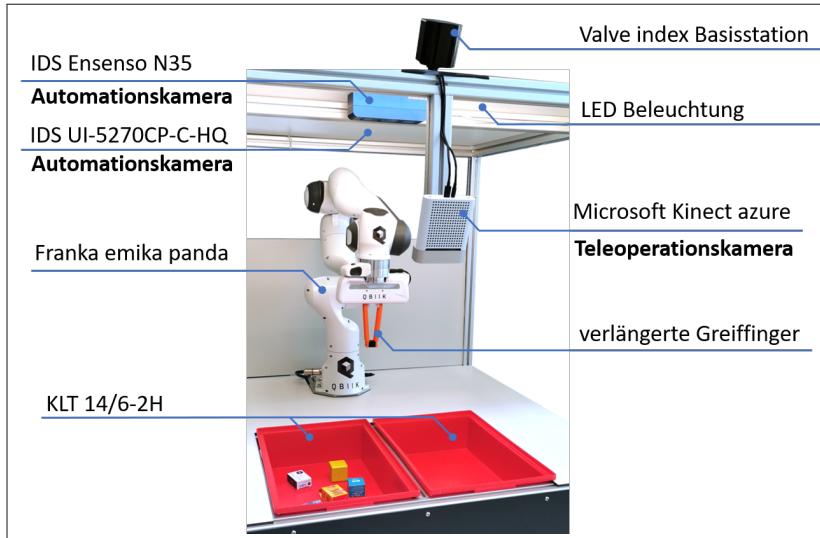


Abbildung 5.1: Entwickelter Demonstrator mit verbauter Sensorik und Aktorik, eigene Darstellung

16BL, [IDS(2022a)]) gehört zur Gruppe der projizierenden Stereokameras und verfügt entsprechend über einen Musterprojektor und zwei Monokameras mit einer Auflösung von 1.3 Megapixel (1280x1024). Beide Kameras wurden so angeordnet, dass beide verbauten KLTs vollständig abgedeckt werden. Durch eine extrinsische Kalibrierung beider Kameras zueinander, kann die Kombination aus Farb- und Tiefenbildkamera als eine RGBD-Kamera, bzw. Automationskamera, betrachtet werden. Beide Kameras werden über PoE-GigE (Power over Ethernet, Gigabit Ethernet) angesteuert. Die Automationskamera liefert Farb- und Tiefenbilder mit einer sehr hohen Auflösung, verfügt jedoch nur über eine Bildwiederholrate (bei höchster Qualität) von ca. 0.5Hz und ist aus diesem Grund für eine Livesteuerung ungeeignet.

Die Teloperationskamera (Microsoft Kinect azure) gehört ebenfalls zur Gruppe der projizierenden Tiefenbildkameras, verfügt jedoch im Gegensatz zur Ensenso N35 nicht über ein Stereokamera-Modul sondern über einen Infrarotprojektor und eine Infrarotkamera. Zusätzlich ist eine Farbbildkamera verbaut. Eine Kalibrierung von Farb- und Tiefenbildkamera zueinander ist nicht notwendig. Die Ansteuerung der RGB-D Kamera erfolgt über USB-C. Die Teleoperati-

onskamera verfügt über eine etwas niedrigere Qualität der Bilder [Tölgessy et al.(2021)Tölgessy, Dekan, Chovanec, und Hubinsky], besitzt jedoch eine Bildwiederholrate von $30Hz$ und eignet sich aus diesem Grund für eine entkoppelte Livesteuering des Knickarmroboters.

Die Ansteuerung aller bisher beschriebenen Komponenten wird durch einen herkömmlichen PC mit einem Intel i-7 Prozessor (Gen 8, 4.0 GHz, 6 Kerne), 16GB Arbeitsspeicher und einer Nvidia 2060 RTX GPU realisiert, der nachfolgend als Steuer-PC bezeichnet wird.

Die VR-Komponenten bestehen aus einem HMD (Head-Mounted-Display), zwei VR-Controllern (linke und rechte Hand) sowie einer Basestation, die in der Mitte des Demonstrators montiert wird. Alle VR-Komponenten gehören zu dem frei am Markt erhältlichen Produkts *Valve index*.

Die VR-Komponenten werden an einen weiteren, herkömmlichen PC mit einem Intel i-7 Prozessor (Gen 7, 2.8 GHz, 4 Kerne), 16GB Arbeitsspeicher und einer Nvidia 1070 GTX GPU angeschlossen, der nachfolgend als Teleoperations-PC bezeichnet. Die Netzwerkverbindung beider PCs erfolgt via Gigabit-Ethernet über einen Netzwerkswitch.

5.2 Teleoperationsmodule

5.2.1 Graphische Benutzeroberfläche

Obwohl die graphische Benutzeroberfläche auf den noch vorzustellenden Modulen *Datenaufnahme auf Seite des Teleoperators* (5.2.2) und *Dekompression und Visualisierung* (5.2.5) basiert, wird sie für ein besseres Verständnis der folgenden Module zuerst vorgestellt. Sie besitzt die wichtige Aufgabe, alle für die jeweilige Teleoperationsstufe relevanten Informationen übersichtlich darzustellen und eine Steuerung möglichst einfach und intuitiv zu ermöglichen. Abbildung 5.3 gibt einen Überblick über die entwickelten Benutzeroberflächen. Für ein besseres Verständnis der Pose des Roboters bzw. der Pose des VR-Controllers wurden diese durch Koordinatensysteme in Orange eingezeichnet.

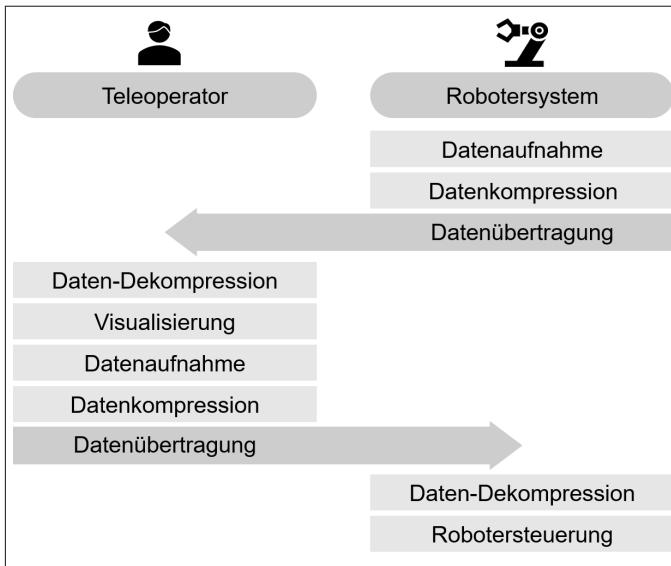


Abbildung 5.2: Schematischer Ablauf der einzelnen Teleoperationsmodule zwischen Teleoperator und Robotiksystem, eigene Darstellung

Für die Teleoperationsstufen 1 und 2, bei denen die Ein- und Ausgangsperipherie lediglich zweidimensional ist, wurden eine Web-Anwendung ((a) in Abbildung 5.3) und eine Smartphone-App ((b) in Abbildung 5.3) entwickelt.

Die Web-Anwendung (Abbildung 5.4 und 5.5) erlaubt einen detaillierteren Einblick in die Aufgabehistorie des CPS. Anzahl an Versuchen, Ergebnis der Objekterkennung und Greifpunktbestimmung, generierte Lerndaten etc. können dem Bedienenden mit wenigen Klicks angezeigt werden. Zusätzlich ist es möglich, neue Objekte einzulernen. Eine Hilfestellung bei der Objekterkennung sowie bei der Greifpunktbestimmung sind, wie bei der Umsetzung in der Smartphone-App, ebenfalls auf einem 2D-Bild durch Boundingbox bzw. Greifpunktpaar möglich. Jede einzelne Stufe des automatisierten Vorganges lässt sich für alle durchgeführten Aufgaben auswählen und das Ergebnis visualisieren.

Die Smartphone-App (Abbildung 5.6) verfügt über eine strukturierte Auflistung von Aufgaben des Teleoperators. Eine dreidimensionale Darstellung der

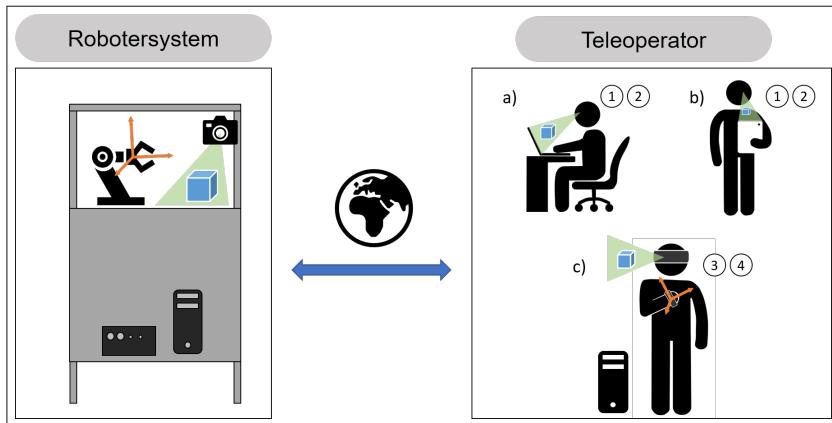


Abbildung 5.3: Drei unterschiedliche graphische Benutzeroberflächen mit ihren Teleoperationsstufen. a) Web-App, Teleoperationsstufen 1 und 2 b) Smartphone/Tablet-App, Teleoperationsstufen 1 und 2 c) Teleoperationssoftware, Teleoperationsstufen 3 und 4, eigene Darstellung

Umgebung (entweder als Karte oder als Live-Punktwolke) sowie eine 2D-Darstellung (als Bild oder Video) sind möglich. Eingaben des Teleoperators werden über den Touchscreen realisiert. Im Falle einer Hilfestellung bei der Objekterkennung kann ein Rahmen (engl. boundingbox) um das zu greifende Objekt gezeichnet und an den Roboter gesendet werden.

Eine direkte Steuerung des Knickarmroboters ist auf Grund der begrenzten Steuermöglichkeiten in zwei Dimensionen weder in der Android-App noch in der Web-Anwendung vorgesehen. Die Hilfestellung des Teleoperators ist somit rein bildbasiert und setzt eine automatisierte Funktion des Systems voraus (Abbildung 4.3). Sollte eine Hilfestellung in den Teleoperationsstufen 1 oder 2 nicht ausreichen, so ist es möglich, direkt aus der Web-Applikation eine Teleoperation in Stufe 3 oder Stufe 4 zu beauftragen.

Für die Teleoperation in Stufe 3 oder Stufe 4 ist die Verwendung eines dedizierten Programmes notwendig ((c) in Abbildung 5.3). Dieses verbindet sich über die in Kapitel 5.2.4 beschriebene Netzwerkschnittstelle mit dem CPS und stellt erhaltene Informationen in der VR-Brille des Teleoperators dar. Der Teleoperator verfügt nun über die Möglichkeit, die vollständige Kontrolle des

5 Umsetzung der Teleoperation

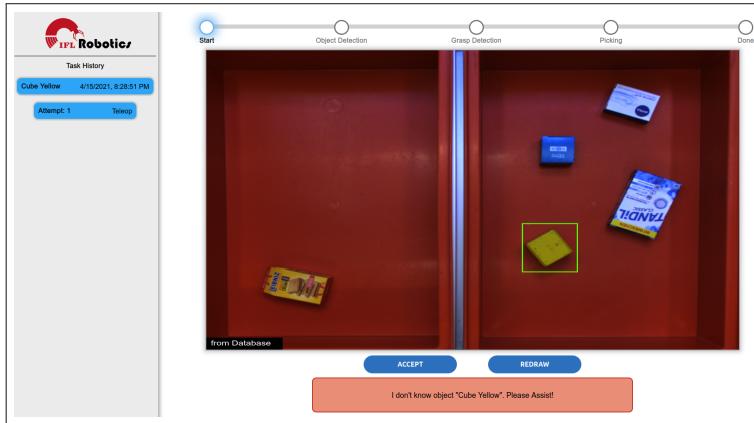


Abbildung 5.4: Web-Applikation für Teleoperationsstufe 1 und 2. Dargestellt ist der Einlernvorgang für ein bisher unbekanntes Objekt. Der Teleoperator kann direkt im Browser eine Boundingbox um das zu greifende Objekt einzeichnen. Eine Hilfestellung der Objekterkennung sieht analog dazu aus, eigene Darstellung

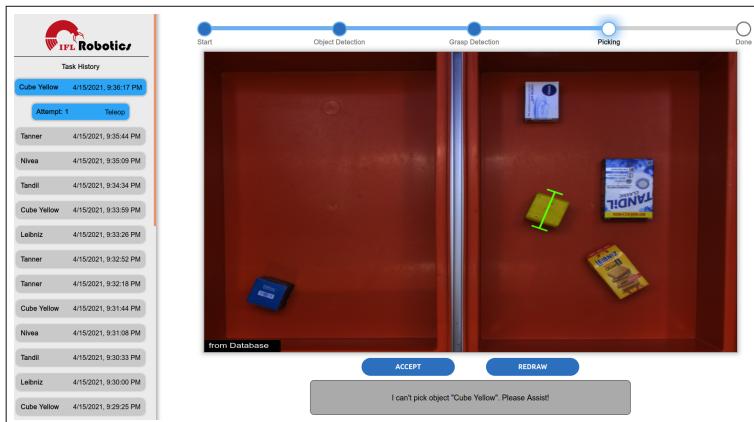


Abbildung 5.5: Darstellung von Teleoperationsstufe 1 und 2 am Beispiel der entwickelten Web-Applikation. Dargestellt ist die Unterstützung beim Greifprozess des Roboters. Der Teleoperator kann direkt im Browser die gewünschten Greifpunkte einzeichnen, eigene Darstellung

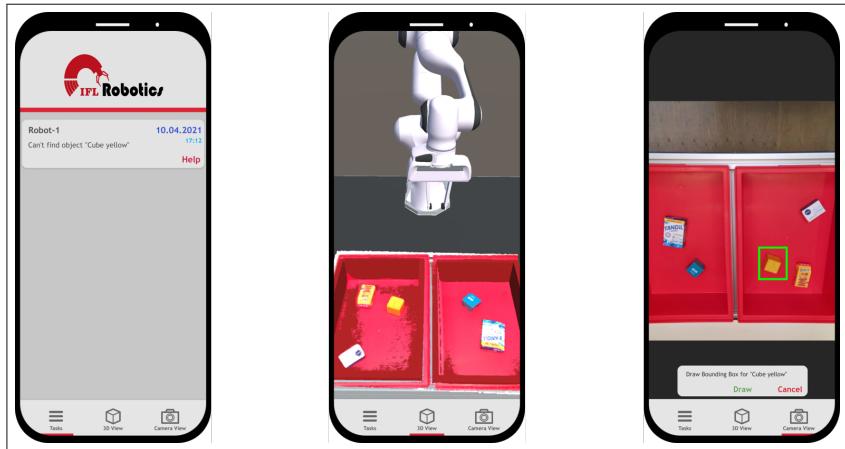


Abbildung 5.6: Darstellung von Teleoperationsstufe 1 und 2 am Beispiel der entwickelten Android-App. Der Teleoperator kann zwischen einer Aufgabenansicht (links), einer 3D-Darstellung (Mitte) und einer 2D-Darstellung wählen, in der über den Touchscreen des Smartphones eine Hilfestellung gegeben werden kann (rechts), eigene Darstellung

Roboters zu übernehmen und durch das überlegene, menschliche Modellwissen die Aufgabe zu lösen (Abbildung 5.7).

Die Android-App sowie das Programm für die Teleoperationsstufen 3/4 wurden auf Basis der *Unity engine* [Technologies(2021)] entwickelt.

5.2.2 Datenaufnahme

Die für die Teleoperation des Knickarmroboters benötigten Daten hängen insbesondere von der gewählten Stufe der Teleoperation (Abbildung 4.3) und somit unmittelbar von der eingesetzten Peripherie ab. Prozessrelevante Daten fallen zum einen auf der Seite des Robotersystems und zum anderen auf der Seite des Teleoperators an. Deren Verknüpfung ist in Abbildung 5.2 dargestellt und wird im folgenden detailliert beschrieben.

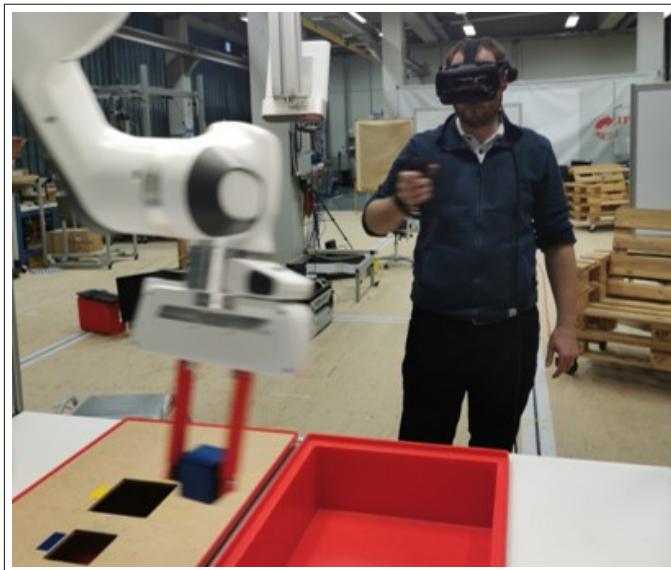


Abbildung 5.7: Teleoperation der Stufe 3/4 mit Visualisierung in VR am entwickelten Demonstrator, eigene Darstellung

Datenaufnahme auf Seite des Roboters Folgende Daten stehen auf Seite des Robotersystems, bedingt durch die verbauten Komponenten des Demonstrators (siehe Kapitel 5.1), zur Verfügung:

- Gelenkwinkelpositionen des Roboters
- Pose des Endeffektors
- Anliegende Momente an den Gelenkwinkeln des Roboters
- Zustand des Greifers
- 2D- und 3D-Bilddaten der Teleoperationskamera
- kolorierte Punktwolke der Teleoperationskamera
- 3D Karte der Umgebung

Der in Eigenleistung entwickelte ROS-Treiber (siehe 5.2.6) stellt die Informationen in Bezug auf den Roboter (Gelenkwinkelpositionen, Gelenkmomente, Pose des Endeffektors) mit einer Frequenz von 100 Hz zur Verfügung. Der Zustand des Greifers, bestehend aus der Position der einzelnen Greiferfinger, wird hardwarebedingt mit einer Frequenz von lediglich 5 Hz bereitgestellt.

Das 2D- und 3D-Bild sowie die kolorierte Punktwolke der Teleoperationskamera werden durch das Softwarepaket *Azure Kinect ROS Driver* [Microsoft(2021a)] unter Verwendung des *Azure Kinect Sensor SDK* [Microsoft(2021b)] bereitgestellt. Das 2D-Farbbild wird in einem Vorverarbeitungsschritt auf Basis der intrinsischen Kameraparameter rektifiziert und in das Bildkoordinatensystem des Tiefenbilds überführt (Abbildung 5.8).

Die extrinsischen Parameter der Teleoperationskamera werden auf Basis des Ansatzes von [Tsai und Lenz(1989)] mit Hilfe der Open-Source Kalibrierungstoolbox *easy handeye* [IFL-CAMP(2021)] berechnet. Für die kamerabasierte Erkennung der Endeffektorpose werden insgesamt vier Aruco-Marker [Garrido-Jurado et al.(2015)], [Romero-Ramirez et al.(2018)] Romero-Ramirez, Muñoz-Salinas, und Medina-Carnicer] quadratisch auf einem Holzbrett positioniert und fest mit dem Roboter verschraubt (Abbildung 5.9).

Die Verwendung von insgesamt vier Aruco-Markern erlaubt eine größere translatorische und rotatorische Bewegung des Endeffektors während des Kalibrierungsprozesses. Zudem wird eine höhere Genauigkeit der kamerabasierten Erkennung der Endeffektorpose erzielt, da die Erkennungsdaten von mehreren erkannten Markern verwendet werden. Die Posen der erkannten Aruco-Marker (rot dargestellt in Abbildung 5.9) werden entsprechend ihrer Position und Orientierung in die Mitte des Brettes verschoben und bilden die Pose des effektiven, kamerabasierten Endeffektorpunktes (grün dargestellt in Abbildung 5.9).

Die 3D-Karte der Umgebung kann entweder als 3D-Standbild oder als eine, während des Betriebs passiv erstellte und sich fortlaufend aktualisierende, 3D-Karte verstanden werden. Für die Generierung einer Umgebungskarte wird auf das Open-Source Paket Octomap [Hornung et al.(2013)] Hornung, Wurm, Bennewitz, Stachniss, und Burgard] zurückgegriffen, das auf Basis der aktuellen Kameraposition und gemessenen Farb- und Tiefendaten eine kolorierte Umgebungskarte erstellt.

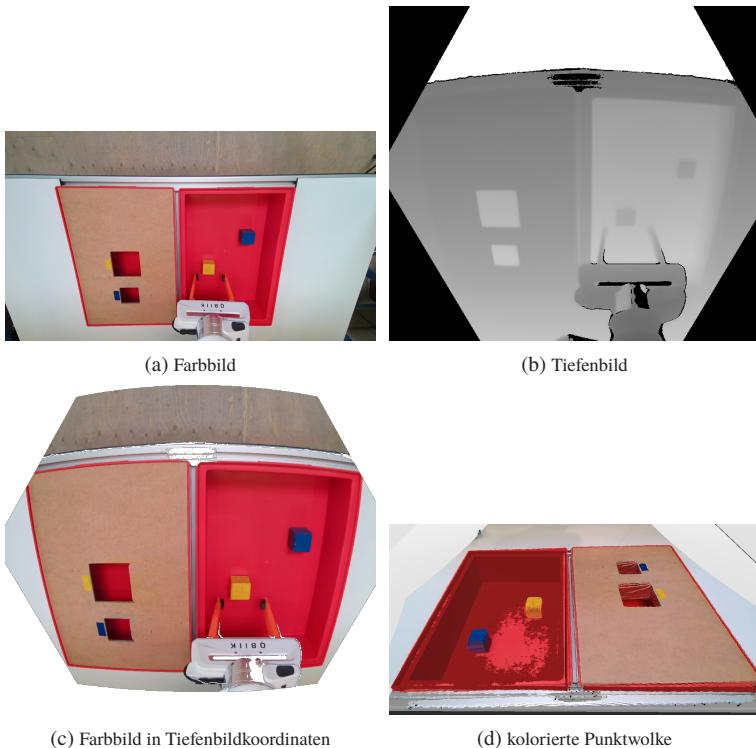


Abbildung 5.8: Beispiel von 2D- und 3D-Bilddaten und kolorierter Punktwolke der Teleoperationskamera. Auf Grund der hohen Auflösung der Teleoperationskamera und einer Darstellung der Punkte durch Polygone wirkt die Punktwolke (trotz anderer Ansicht) wie ein Foto, eigene Darstellung

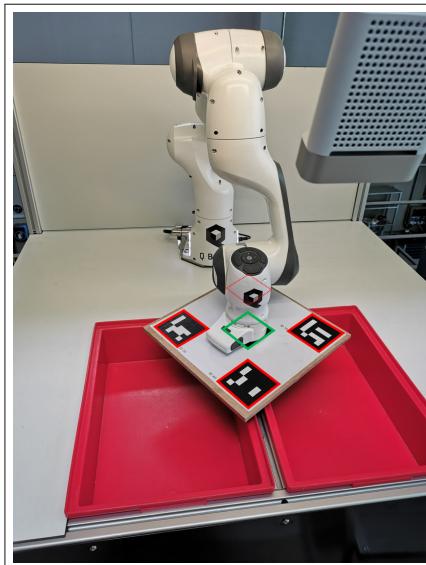


Abbildung 5.9: Extrinsische Kalibrierung der Teleoperationskamera. Erkannte Aruco-Marker werden in rot dargestellt, verdeckter Aruco-Marker mit gestrichelter roter Linie sowie die effektive Endeffektorpose in grün dargestellt, eigene Darstellung

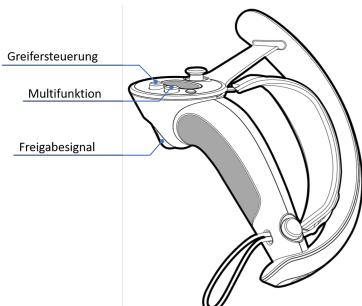
Datenaufnahme auf Seite des Teleoperators Auf Seite des Teleoperators werden je nach Stufe der Teleoperation folgende Daten generiert:

- Eingezeichnete Objektpositionen
- Eingezeichnete Greifpunkte
- Freigabesignal
- Pose der Hand des Teleoperators
- Pose des Kopfes des Teleoperators
- Befehle zum Öffnen/Schließen des Greifers
- Benutzerspezifische Befehle

Die Steuerungsmöglichkeiten sind für die Teleoperationsstufen 1 und 2 auf Grund der eingesetzten Peripherie stark limitiert. Dem Teleoperator stehen lediglich Peripheriegeräte zur Verfügung, die Daten in zwei Dimensionen aufnehmen und darstellen können. Aus diesem Grund ist eine sichere, direkte Steuerung eines komplexen Robotersystems ausgeschlossen. Durch die Aufnahme von *Objektpositionen* und möglichen *Greifpunkten* können dem Robotersystem dennoch wichtige Handlungsempfehlungen gegeben werden, die ein Ausfallen oder ein Fehlverhalten des Systems verhindern.

Für die Teleoperationsstufen 3 und 4 steht dem Teleoperator ein deutlich umfangreicheres Spektrum an Steuersignalen zur Verfügung. Diese bestehen aus einem *Freigabesignal* (ähnlich einem Totmannschalter), einem Befehl zum *Öffnen oder Schließen des Greifers* sowie die aktuelle Position und Orientierung von *Hand* und *Kopf* des Teloperators. Eine Übertragung von *benutzer-spezifischen Befehlen* ist durch eine Multifunktions-taste vorhergesehen. Die Knopfbelegungen sind in Abbildung 5.10 dargestellt.

Die Pose der Hand des Teleoperators wird im Folgenden als p_t bezeichnet und mit Hilfe eines VR-Controllers ausgelesen. Diese besteht aus sieben Werten: die Position $p_{tp} = (x, y, z)$ sowie die Rotation $p_{tr} = (s, q_1, q_2, q_3)$ als Quaternion (siehe 3.1.2).



(a) Knopfbelegungen VR-Steuerung



(b) Mensch steuert Knickarmroboter

Abbildung 5.10: Steuerung des Knickarmroboters mit Hilfe eines VR-Controllers. Die Pose der Hand des Teleoperators wird unmittelbar an den Endeffektor des Roboters übertragen. Die vollständige Steuerung des Roboters benötigt nur drei Knöpfe, eigene Darstellung

Damit dem Teleoperator die Möglichkeit gegeben wird die Steuerungen an einem beliebigen Punkt im Raum zu beginnen bzw. zu beenden, wird eine weitere virtuelle Pose p_v (analog zu p_t ausgedrückt durch p_{v_p} und p_{v_r}) eingeführt. Zu Beginn der Steuerung wird die virtuelle Pose p_v an die reale Pose des Endeffektors p_{ee} des Knickarmroboters angeglichen. Anschließend wird die Differenz zwischen virtueller Pose p_v und tatsächlich gemessener Eingangspose p_t verglichen und somit neben der relativen Distanz r_d auch die relative Orientierung r_o bestimmt:

$$r_d = p_{t_p} - p_{v_p} \quad (5.1)$$

$$r_o = p_{t_r}^{-1} \cdot p_{v_r} \quad (5.2)$$

Führt der Teleoperator eine Bewegung aus, wird die um die relative Bewegung r_d veränderte virtuelle Pose p_v mit

$$p_{v_p} = p_{t_p} - r_d \quad (5.3)$$

$$p_{v_r} = p_{t_r} \cdot r_o \quad (5.4)$$

aufgezeichnet. Anfänglich wurde lediglich die Differenz zwischen p_t und p_v unmittelbar an die geschwindigkeitsbasierte Regelung des Roboters in 5.2.6 übertragen. Das hat allerdings bei einem verzögerten Steuersignal, auf Grund einer zu hohen bzw. variablen Latenz, zu einem Aufschwingen des Knickarmroboters geführt. Die hier vorgestellte Berechnung in Relation zur letzten erhaltenen Endeffektor Position p_{ee} beugt einem Aufschwingen bei zeitlich verzögerten Signalen durch die Übertragung einer absoluten Soll-Position vor.

Die Pose des Kopfes des Teleoperators p_{kt} wird nicht relativ zur Endeffektorposition des Roboters bestimmt und entsprechend als Absolutwert aufgenommen.

5.2.3 Datenkompression

Die Wahl einer geeigneten Datenkompression ist für die Qualität der Teleoperation von entscheidender Bedeutung und abhängig von der jeweiligen Teleoperationsstufe. Nachfolgend soll zwischen der Datenkompression für eine zeitverzögerte Kommunikation und einer Echtzeit-Kommunikation unterschieden werden. Die zu komprimierenden Daten lassen sich in beiden Fällen in ein-

oder mehrdimensionale Felder (wie z.B. Gelenkwinkel des Knickarmroboters), Bilddaten (Farb- und Tiefenbild) und 3D-Daten (Punktwolken) einteilen.

Die Kompression bzw. Serialisierung der ein- oder mehrdimensionalen Vektorfelder wird in beiden Varianten durch *Google FlatBuffers* [Google(2021)] durchgeführt.

Datenkompression bei zeitverzögterer Kommunikation Für eine zeitverzögerte Kommunikation zwischen CPS und Teleoperator, wie sie für die Teleoperationsstufe 1 und 3 vorgesehen ist, spielt die Dauer des Kompressions- und Dekompressionsvorgangs sowie die Größe der zu übertragenden Daten eine untergeordnete Rolle. Der Teleoperator trifft seine Entscheidung lediglich auf Basis eines Schnappschusses der Vergangenheit, eine Dauer von mehreren Sekunden für Kompression, Übertragung und Dekompression beeinträchtigt die Funktionalität nicht merklich. Vielmehr steht eine möglichst detailgetreue Darstellung der Umgebung und eine einfache Handhabung der Daten im Vordergrund.

Für die Kompression von Bilddaten wird aus diesem Grund auf die gängige *Portable-Network-Graphics*-Komprimierung oder kurz *png-Komprimierung* zurückgegriffen. Sie gehört zur Gruppe der *verlustfreien Kompressionsverfahren* (siehe Kapitel 3.3.1). Die Verwendung von *libpng* [libpng(2021)] ist bereits seit vielen Jahren Stand der Technik. Die png-Komprimierung ist für die Anwendung der Teleoperationsstufe 1 eines Kommissionierroboters ausreichend. An dieser Stelle sei jedoch der neue *JPEG XL*-Standard [Alakuijala et al.(2019)Alakuijala, Asseldonk, und Boukortt et al.] erwähnt, der in Zukunft möglicherweise eine relevante Alternative darstellen könnte [Kliuchnikov und Upenik et al.(2020)].

Eine effiziente Komprimierung von 3D-Punktwolken ist erst seit einigen Jahren aktueller Bestandteil der Forschung [Cao et al.(2019)Cao, Preda, und Zaharia]. Hierbei wird insbesondere zwischen zwei möglichen Verfahren unterschieden: dem videobasierten- und dem geometriebasierten Kompressionsverfahren. Beide Verfahrenskategorien werden aktuell als ISO/IEC 23090-5 [ISO(2021)] bzw. ISO/IEC JTC 1/SC 29 [ISO(2021)] standardisiert.

Die videobasierten Verfahren basieren auf Kompressionsverfahren aus dem Bereich der Videokompression und werden im folgenden Kapitel 5.2.3 detailliert vorgestellt.

Geometriebasierte Verfahren suchen Muster innerhalb der 3D-Daten und leiten Regeln zur Rekonstruktion ab. Beide Verfahren können entweder als verlustfreie (siehe Kapitel 3.3.1) oder als verlustbehaftete (siehe Kapitel Quantisierung 3.3.2) Kompressionsverfahren realisiert werden.

Wie bereits in Kapitel 5.2.2 beschrieben, wird in der vorliegenden Arbeit für die Generierung der 3D-Karte auf das Paket *Octomap* [Hornung et al.(2013)Hornung, Wurm, Bennewitz, Stachniss, und Burgard] zurückgegriffen, dass die vorliegende Punktwolke in unterschiedlichen Detaillierungsebenen speichert und anschließend entsprechend der gewünschten Detaillierungsebene aufgelöst werden kann. Eine weitere Kompression vor der Datenübertragung wurde nicht durchgeführt.

Datenkompression bei Echtzeit-Kommunikation Im Gegensatz zur Datenkompression für eine zeitverzögerte Kommunikation (siehe 5.2.3) steigen die Anforderungen an die Kompressionsalgorithmen für eine Echtzeitkommunikation, wie sie in den Teleoperationsstufen 2 und 4 benötigt werden. Das gilt sowohl für die Dauer des Kompressions- und Dekompressionsvorgangs als auch für die tatsächliche Größe der zu übertragenen Daten. Weiterhin soll eine bestmögliche Qualität der Daten gewährleistet sein.

Da es sich bei der Übertragung von Bilddaten nicht nur um einzelne Bilder, sondern um einen Stream, also viele aufeinander folgende und voneinander abhängige Bilder handelt, können Kompressionsalgorithmen verwendet werden, die eine zeitliche Abfolge berücksichtigen. Des Weiteren ist es auf Grund von limitierten Netzwerkressourcen, wie z.B. dem Netzwerkdurchsatz (siehe Kapitel 3.2.2) oder der Latenz (siehe Kapitel 3.2.2) notwendig, auf die Gruppe der verlustbehafteten Kompressionsverfahren (siehe Kapitel 3.3.2) zurückzugreifen.

Folgende Tabelle gibt eine Übersicht über die zu komprimierenden Daten der Teleoperationskamera (vgl. 5.1):

Die Datenmenge von Farbbild und Tiefenbild berechnet sich aus der Anzahl an Pixeln sowie der zu speichernden Datenmenge pro Pixel. Das Farbbild besitzt

Name	Auflösung	Hz	einzeln	Stream
Farbbild ¹	1280x720	30	2.76 MB	82.94 MB/s
Farbbild ²	640x576	30	1.11 MB	33.2 MB/s
Tiefenbild	640x576	30	0.74 MB	22.1 MB/s
kol. Punktwolke	640x576	30	5.53 MB	165.9 MB/s

Tabelle 5.1: Übersicht über die Datenmenge von Farbbild, Tiefenbild und Pointcloud bei einer Aktualisierungsrate von 30 Hz. (1) Farbbild in HD Auflösung (2) Farbbild projiziert in Tiefenbildkoordinaten

drei Kanäle (Rot, Grün, Blau) die jeweils durch ein 8-Bit Integer (0-255) beschrieben werden. Das Tiefenbild besitzt nur einen Kanal und wird durch ein 16-Bit Integer (0-65535) beschrieben. Die kolorierte Punktwolke wird aus den Informationen von Farbbild und Tiefenbild berechnet. Jeder 3D-Punkt wird durch den Farbwert (RGB = 24-Bit) sowie der dreidimensionalen Position im Raum (x,y,z) durch drei Float-Werte mit jeweils 32-Bit beschrieben. Die zu speichernden Informationen einer kolorierten Punktwolke sind mit 120-Bit pro Punkt also deutlich größer als von Farbbild und Tiefenbild zusammen.

Je nach Stufe der Teleoperation wird entweder nur das Farbbild oder eine kolorierte Punktwolke auf Seite des Teleoperators benötigt. Findet die Berechnung der Punktwolke auf Seite des Teleoperators statt, kann je nach Stufe der Teleoperation das Tiefenbild entweder gesendet oder weggelassen werden. Aus diesem Grund wurde sich für die Verwendung einer videobasierten Kompression für die Punktwolke entschieden.

Die zwei gängigsten Standards bei der Videokompression sind *Advanced Video Coding*, auch bekannt als *H.264* sowie dessen Nachfolger *High Efficiency Video Coding*, bekannt als *H.265* oder *HEVC*. Beide Standards liegen in einer Quelloffenen Implementierung vor [Cisco(2021)] [StrukturAG(2021)]. Bei der Einbindung der beiden Kompressionsstandards in das Teleoperations-framework gab es zwei grundsätzliche Herausforderungen:

- Verlustbehaftete Kompression
- CPU-Auslastung des CPS

auf die im folgenden genauer eingegangen wird.

Verlustbehaftete Kompression Während eine verlustbehaftete Kompression von Farbbildern überwiegend auf einer Irrelevanzreduktion gewisser Farbinformationen beruht (Chromo-Subsampling 3.3.2), lässt sich dieses Verfahren nicht auf ein Tiefenbild übertragen. Die Tiefeninformationen werden, ähnlich wie bei einem Farbbild, durch einen Helligkeitswert angegeben. Allerdings lassen sich hierbei keine Effekte der menschlichen Perzeption für eine Irrelevanzreduktion ausnutzen. Jeder Helligkeitswert des Tiefenbildes entspricht exakt einer Position im dreidimensionalen Raum. Eine Quantisierung würde zu einer verfälschten Darstellung in der berechneten Punktwolke auf Seite des Teleoperators führen. Aus diesem Grund wird für das vorliegende Farbbild eine verlustbehaftete Kompression gewählt und für das Tiefenbild eine verlustfreie Kompression.

CPU-Auslastung des CPS Die sequenzielle Kompression von Farb- und Tiefenbild mit *OpenH264* [Cisco(2021)] sowie mit *libde265* [StrukturAG(2021)] führt zu einer hohen Auslastung der CPU des CPS. Eine Echtzeitkompression ist mit dem verwendeten System (vgl. Kapitel 5.1) zwar möglich, allerdings ist eine Einschränkung der anderen benötigten Module, wie der Datenaufnahme, der Datenübertragung sowie der zeitkritischen Robotersteuerung, nicht auszuschließen. Da das CPS über eine moderne GPU verfügt, ist eine Auslagerung des Kompressionsvorgangs auf die GPU sinnvoll. Das *Nvidia Video SDK* [NVIDIA(2013)] stellt hierfür eine Schnittstelle für eine hardwarebeschleunigte Kompression von Bilddaten für die Kompressionsstandards *H.264* und *H.265* ab der *Maxwell Architektur* [NVIDIA(2013)] bereit, die unabhängig von den vorhandenen CUDA-Kernen arbeitet. Die verbaute GPU verfügt über zwei hardwarebasierte De-/Kompressionsmodule, die als *NVDecoder* sowie *NVEncoder* bezeichnet werden und die NV12-Kodierung (siehe Kapitel 3.3.2) verwenden. Die hardwarebeschleunigte Kompression auf der GPU ermöglicht es nicht nur eine 6x schnellere Kompression als auf der CPU zu erzielen, die Kompression findet auch ohne die Verwendung von Ressourcen statt, die möglicherweise an anderer Stelle benötigt werden. Zusätzlich erlaubt die GPU-basierte Kompression von Farb- und Tiefenbild einen erheblichen Vorteil bei der Visualisierung der kolorierten Punktwolke, der in Kapitel 5.2.5 vorgestellt wird.

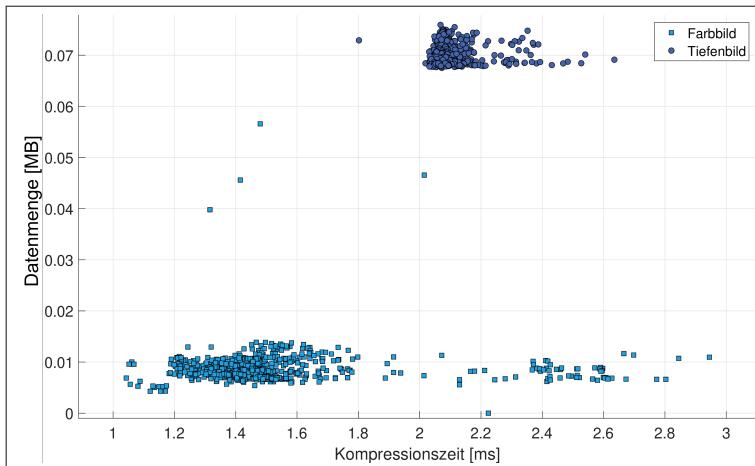


Abbildung 5.11: Vergleich von Kompressionsgröße und Kompressionsdauer von Farb- und Tiefenbild während des Teleoperationsvorgangs. Die Kompressionszeit beinhaltet die Übertragung des Bildes zur GPU, die Kompression auf der GPU sowie den Kopiervorgang des komprimierten Bildes zurück zur CPU, eigene Darstellung

Abbildung 5.11 zeigt den Zusammenhang von Kompressionszeit und Kompressionsgröße von Farb- und Tiefenbild während eines vollständigen Teleoperationsprozesses. Die Kompressionszeit beinhaltet die Übertragung des Bildes zur GPU, die Kompression auf der GPU sowie den Kopiervorgang des komprimierten Bildes zurück zum Speicher der CPU (Arbeitsspeicher). Die angegebene Datenmenge entspricht einem zu übertragenden Bild bzw. Frames 3.3.2. Die durchschnittliche Datenmenge für das mit einer verlustbehafteten Kompression komprimierte Farbbild beträgt 0.0087 MB. Das verlustfrei komprimierte Tiefenbild besitzt noch eine Größe von durchschnittlich 0.07 MB.

5.2.4 Datenübertragung

Die Datenübertragung zwischen Robotersystem und Teleoperator findet über eine Netzwerkverbindung statt. Entsprechend der Netzwerkschichten (siehe Kapitel 3.2.1), kann als *Netzzugang* entweder Ethernet oder WLAN verwendet werden.

Name	unkomprimiert	komprimiert ¹	Faktor ¹
Farbbild	1.11 MB	0.0087 MB	1 : 127
Tiefenbild	0.74 MB	0.07 MB	1 : 10
kol. Punktwolke	5.53 MB	0.079 MB ²	1 : 70 ²

Tabelle 5.2: Kompressionsfaktor von Farb- und Tiefenbild sowie der kolorierten Punktwolke. (1) Die angegebenen Werte beziehen sich auf die durchschnittliche Datenmenge während einer Teleoperation. (2) Die kolorierte Punktwolke wird aus komprimierten Farb- und Tiefenbild berechnet, die komprimierte Größe entspricht somit der Summe der komprimierten Bilddaten.

Die *Vermittlungsschicht* basiert auf IPv6. Da bei der Teleoperation die Übertragung besonders zeitkritisch ist, wurde sich für das UDP-Protokoll (siehe 3.2.3) in der *Transportschicht* entschieden. Dieses besitzt im Gegensatz zu dem TCP-Protokoll (siehe 3.2.3) einen kleineren Header und keine *Sicherungsschicht*. Das Wegfallen der Sicherungsschicht hat zum einen den Vorteil, das weitere Bandbreite gespart werden kann. Zum anderen wird ein Aufstauen von gesendeten Paketen verhindert, da keine Empfangsbestätigungen (ACK) verschickt werden und kein erneutes Senden von verlorenen Paketen (siehe 3.2.2) stattfindet.

Werden Sequenznummer und Anzahl der gesendeten Pakete zwischen Sender und Empfänger synchronisiert, lässt sich mit UDP eine sichere und vor allem echtzeittaugliche Netzwerkverbindung herstellen.

Bevor die ausgehenden Pakete an die *Vermittlungsschicht* übergeben oder die eingehenden Pakete weiterverarbeitet werden, findet eine Zwischenspeicherung in einem *Ringbuffer* [Chandrasekaran(2014)] statt (Abbildung 5.12). Die *Ringbuffer* werden sowohl Server- als auch Clientseitig eingesetzt. Sie ermöglichen einen *threadsicheren* Lese- und Schreibzugriff nach dem FiFo-Prinzip (First In - First Out).

Abbildung 5.13 visualisiert alle ausgehenden Daten sowie die CPU-Auslastung des CPS während einer aktiven Stufe 4 Teleoperation. In Bereichen die mit 1 gekennzeichnet sind, findet lediglich die Übertragung der Umgebungsdaten in Form von Farb- und Tiefenbild statt. Die durchschnittlich benötigte Bandbreite für das Farbbild beträgt 0.26 MB/s sowie für das Tiefenbild 2.1 MB/s.

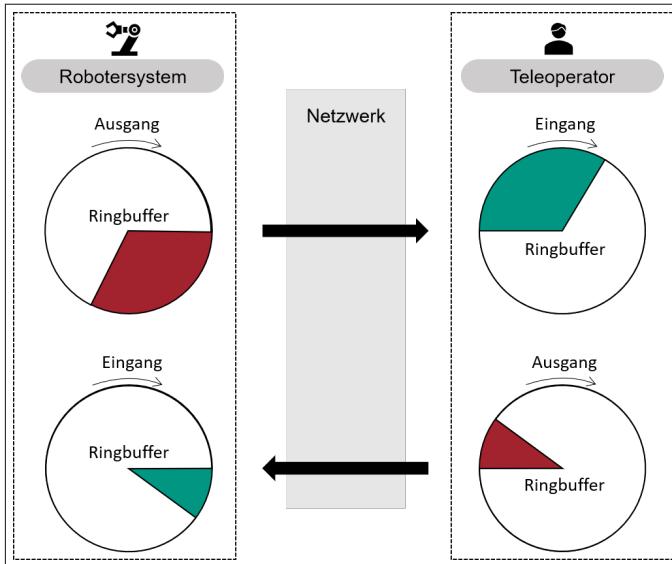


Abbildung 5.12: Eingangs- und Ausgangs Ringbuffer als FiFo-Prinzip bei der Datenübertragung, eigene Darstellung

In Bereichen die mit 2 gekennzeichnet sind ist eine höhere Netzwerkauslastung erkennbar. Das ist zum einen durch die interne Kommunikation zwischen Steuer-PC und Knickarmroboter zurückzuführen, bei der diverse Befehle mit einer Frequenz von 1000 Hz ausgetauscht werden. Zum anderen ist auch bei Farb- und Tiefenbild eine etwas höhere Netzwerklast zu beobachten. Diese lässt sich durch die Bewegung des Knickarmroboters erklären, die zu einer häufigeren Übertragung von I-Frames (siehe Kapitel 3.3.1) führt. Insgesamt liegt die benötigte Bandbreite für die Übertragung der gesamten Visualisierung während der Teleoperation durchschnittlich bei 2.36 MB/s bzw. 18.88 Mbit/s und entspricht somit der mindestens notwendigen, durchschnittlichen Nettoübertragungsrate (siehe 3.2.2) zwischen Robotersystem und Teleoperator. Ein Einsatz über WLAN ist somit theoretisch ab dem Standard 802.11a/g/h (siehe Tabelle 3.2) möglich.

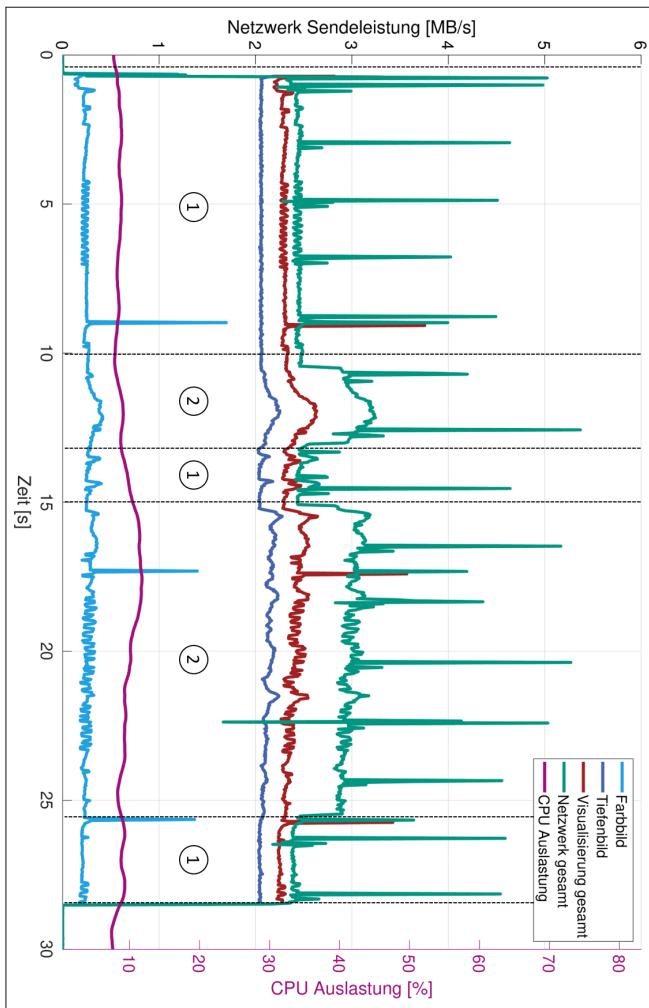


Abbildung 5.13: Netzwerk- und CPU-Auslastung während des Teleoperationsvorgangs. In Bereichen, die mit 1 kennzeichnet sind, findet lediglich die Übertragung der Umgebungsdaten in Form von Farb- und Tiefenbild statt. Die durchschnittlich benötigte Bandbreite für das Farbbild beträgt 0.26 MB/s sowie für das Tiefenbild 2.1 MB/s. Bereiche, die mit 2 gekennzeichnet sind, beinhalten zusätzlich die Steuerung des Knickarmroboters. Die benötigte Bandbreite für die Übertragung der gesamten Visualisierung liegt durchschnittlich bei 2.36 MB/s, eigene Darstellung

5.2.5 Dekompression und Visualisierung

Aufgabe der Dekompression und Visualisierung ist eine möglichst verzögerungsfreie Darstellung der empfangenen Daten mit Hilfe der verfügbaren Hardware und *Ausgabe-Peripherie* auf Seite des Teleoperators. Das gilt für verschiedene Sensordaten des Knickarmroboters, für das Farb- und Tiefenbild und die kolorierte Punktwolke der Teleoperationskamera.

Für die Darstellung der aktuellen Roboterpose werden die Gelenkwinkelpositionen des Knickarmroboters sowie der aktuelle Zustand des Greifers verwendet. Die Darstellung findet auf einem möglichst realitätsnahen digitalen Zwilling des Robotersystems statt (Abbildung 5.14). Bei der Umsetzung wurde auf eine optimierte Geometrie (wenig Polygone) geachtet, die ein schnelles Rendering und somit eine Darstellung in Echtzeit ermöglicht.

Die dreidimensionale Darstellung des digitalen Zwilling findet in allen Stufen der Teleoperation Verwendung:

Für Stufe 1 und Stufe 2, bei der die Hilfestellung überwiegend auf der Auswertung von 2D-Bilddaten durch den Teleoperator basiert, unterstützt die aktuelle Pose des Knickarmroboters das Szenenverständnis. Des Weiteren lassen sich Bewegungsabläufe, z.B. von Fehlversuchen, intuitiv darstellen und durch den Teleoperator entsprechend korrigieren.

Ein weiter Vorteil ist die direkte Verbindung zu den aufgenommenen 3D-Daten am Robotersystem. Die Darstellung einer 3D-Karte lässt sich problemlos mit der Darstellung des digitalen Zwilling kombinieren.

Verfügt der Teleoperator über dreidimensionale Ein- und Ausgabe-Peripherie (Teleoperationsstufe 3 und 4), führt die Kombination von statischen 3D-Modellen und dynamischen 3D-Daten zu einem intuitiven Umgebungsverständnis und einem hohen Immersionsgrad (siehe 7.2).

Im Rahmen dieser Arbeit wurden die Module *CUDA Point Cloud Calculator (PCC)* und *Pointcloud Shader (PC Shader)* entwickelt, die eine effiziente Berechnung und Visualisierung der kolorierten Punktwolke auf der GPU des Teleoperators durchführen. Die Implementierung von *CUDA PCC* basiert auf dem *Azure Kinect Sensor SDK* [Microsoft(2021b)] und *k4a cuda* [conatsera(2019)]. Die mathematische Grundlage für *CUDA PCC* wird in den Gleichung 5.5-5.8 beschrieben.

Die Implementierung des *PC Shader* greift direkt auf den Speicherbereich von *CUDA PCC* zu und basiert auf *Pcx* [Takahashi(2021)]. Die Struktur von Datenaufnahme, Datenkompression, Datenübertragung, Dekompression und Visualisierung mit *CUDA PCC* und *PC Shader* ist in Abbildung 5.15 dargestellt.

Der Vorteil in der Verwendung von *CUDA PCC* und *PC Shader* ist zum einen die hohe Parallelisierbarkeit der Punktwolkenberechnung. Die Ausnutzung der GPU bietet eine effiziente Berechnungsmöglichkeit der Punktwolke durch parallele Berechnung. Zum anderen wird ein wiederholtes Kopieren der 3D-Daten zwischen dem Speicher von GPU und CPU unnötig. Wie in Abbildung 5.15 zu erkennen kann die kolorierte Punktwolke direkt von der GPU auf der Ausgabe-Peripherie des Teleoperators angezeigt werden. Die Verknüpfung von GPU-basierter Dekompression von Farb- und Tiefenbild, GPU-basierter Punktwolkenberechnung und Übergabe an den *PC Shader* erlaubt die Verwendung des dedizierten GPU-Speichers für alle beteiligten Module.

Für die parallele Berechnung während der Laufzeit wird eine zweidimensionale Wertetabelle $w_{u,v}$ (engl. Lookup-Table) im dedizierten GPU-Speicher generiert. Jeder Wert beschreibt hierbei die Transformation bzw. Projektion eines Pixels aus dem Tiefenbild und dem zugehörigen Pixel aus dem Farbbild in die kolorierte Punktwolke in Weltkoordinaten.

Basierend auf der inversen Kameramatrix \mathbf{K}^{-1} aus Gleichung 3.32 des Lochkameramodells mit $\theta = \pi/2$ erhält man die Projektionsvorschrift

$$\mathbf{x} = \mathbf{K}^{-1} \cdot \mathbf{p} \cdot \mathbf{z}^T \quad (5.5)$$

mit den Bildkoordinaten \mathbf{p} und zugehörigen Tiefenwerten \mathbf{z} zu den Kamerakoordinaten \mathbf{x} . Somit ergibt sich durch weglassen von \mathbf{z}^T und hinzunehmen von B_x und B_y , die eine radiale und tangentiale Verzerrungskorrektur nach dem *Brown-Conrady Modell* [Brown(1966)], [Conrady(1919)] darstellen, die Wertetabelle $w_{u,v}$. Eine notwendige Rektifizierung wird durch B_x und B_y direkt in die Projektion durch die Wertetabelle integriert:



Abbildung 5.14: Digitaler Zwilling des Demonstrators mit optimierter Geometrie für ein schnelles Rendering. Darstellung ohne verbaute Sensorik. Gelenkwinkel, Zustand des Greifers sowie kolorierte Punktwolke können interaktiv und in Echtzeit visualisiert werden, eigene Darstellung

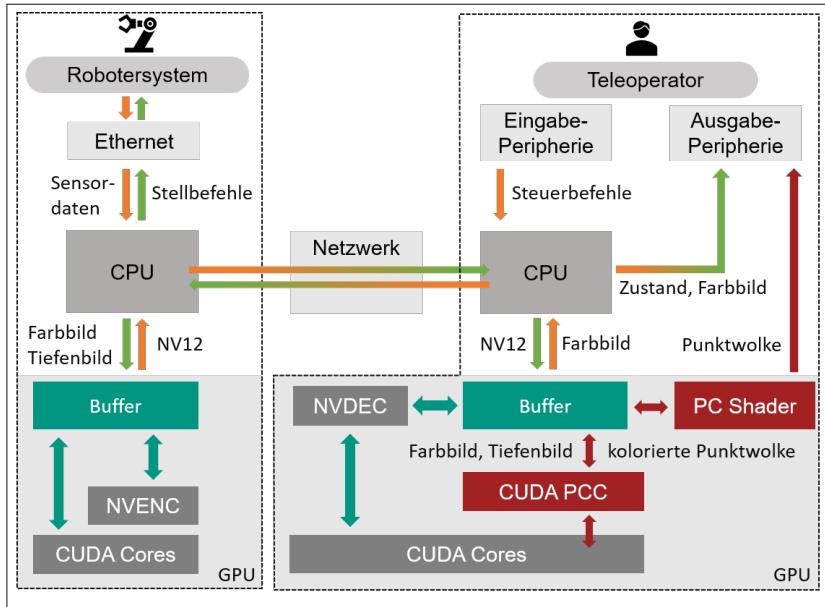


Abbildung 5.15: Struktur von Datenaufnahme, Datenkompression, Datenübertragung und Visualisierung mit *CUDA PCC* und *PCC Shader* zwischen Teleoperator und Robotiksystem, eigene Darstellung

$$w_u = \frac{u - u_0}{fx} \cdot B_x \quad (5.6)$$

$$w_v = \frac{v - v_0}{fy} \cdot B_y \quad (5.7)$$

Die Punktfolke kann anschließend parallelisiert durch Multiplikation jedes Pixels des Tiefenbildes mit der Wertetabelle $w_{u,v}$ berechnet werden:

$$\mathbf{P}_{x,y,z} = \underbrace{p(u, v) \cdot w_{u,v}}_{= z}, \quad \forall p \in \mathbf{p} \quad (5.8)$$

und mit der Farbinformation des Farbbildes ergänzt werden.

Unter Verwendung der vorgestellten Berechnungsmethodik beträgt die Dekompressionsdauer im Durchschnitt 2.48 ms für das Farbbild und 2.37 ms für das Tiefenbild. Die anschließende Berechnung der kolorierten Punktwolke, inklusive Visualisierung benötigt im Schnitt 0.025 ms. Die gemessenen Zeiten wurden in über 1400 Bildern während einer Teleoperation gemessen und sind in Abbildung 5.16 dargestellt. Die Gesamtdauer für Dekompression und Visualisierung der vollständigen Punktwolke addiert sich somit auf $2.48ms + 2.37ms + 0.025ms = 4.875ms$.

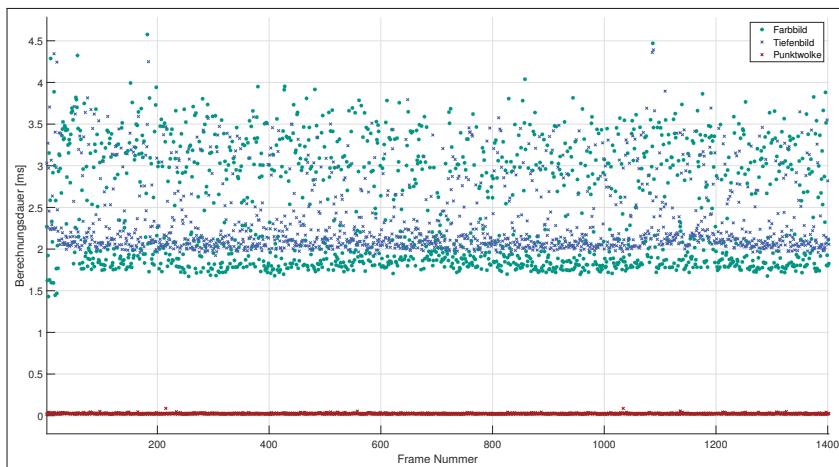


Abbildung 5.16: Vergleich der Dekompressionsdauer von Farb- und Tiefenbild sowie der Berechnung der kolorierten Punktwolke, eigene Darstellung

5.2.6 Robotersteuerung

Die Steuerung des verwendeten Knickarmroboters *Franka emika panda* (siehe 5.1) erfolgt mit Hilfe eines selbstentwickelten c++-Treivers für ROS, das für die Kommunikation mit dem Roboter das Franka-Control-Interface (FCI) ([Emika(2021)]) verwendet (Abbildung 5.17).

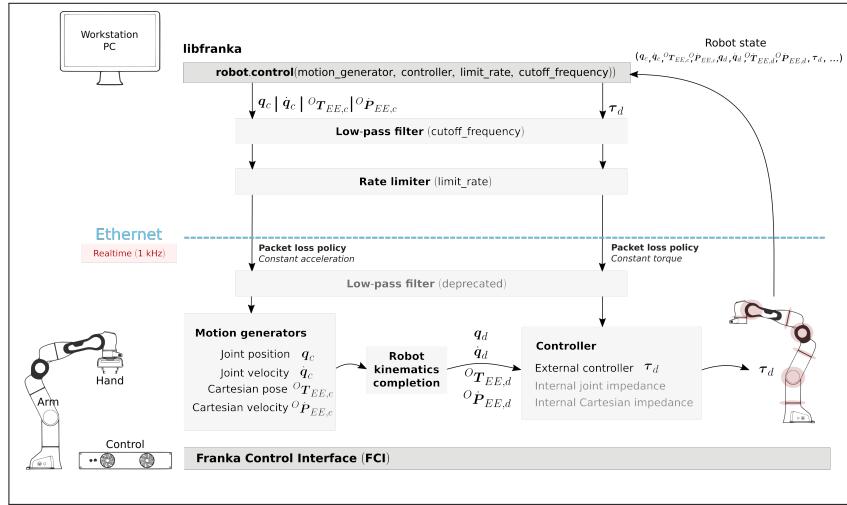


Abbildung 5.17: Schematische Darstellung des Franka-Control-Interface (FCI) [Emika(2021)]

Das FCI stellt eine Echtzeitkommunikation mit einer Frequenz von 1000 Hz via Ethernet zum Knickarmroboter her. Wie in Abbildung 5.17 dargestellt, wird der aktuelle Zustand der Roboters *robot state* ausgelesen und über das Modul *Motion generators* gewünschte Stellbefehle an den Roboter übertragen. Das eigens entwickelte Interface bietet die Möglichkeit eines schnellen Wechsels zwischen den einzelnen Bewegungsreglern (engl. *Motion generators*) und zudem eine einfache Fehlerbehebungsroutine im Falle einer Kollision. Das entwickelte Interface ermöglicht einen fließenden Wechsel zwischen automatisierter und manueller Steuerung des Roboters.

Nachfolgend wird eine Iterationsschleife der Roboterregelung unter Verwendung des kartesischen Geschwindigkeitsreglers (engl. *Cartesian velocity motion generator*) beschrieben.

Zu Beginn wird die Transformationsmatrix T_{O_EE} , also die Transformationsvorschrift von der Roboterbasis O zum Endeffektor EE ausgelesen und in die Pose p_{ee} mit den Anteilen p_{ee_p} für die kartesische Position und p_{ee_r} als Rotation des Endeffektors in Form eines Quaternionen überführt. Anschließend wird deren Differenz $d\vec{p}$ und $d\vec{r}$ zwischen der durch den Teleoperator generierten Sollpose p_v (siehe 5.2.2) und aktueller Endeffektor Position p_{ee} gebildet:

$$d\vec{p} = p_{ee_p} - p_{v_p} \quad (5.9)$$

$$d\vec{r}_Q = p_{v_r} \cdot p_{ee_r}^{-1} \quad (5.10)$$

Da die kartesische Geschwindigkeitsregelung nicht mit Quaternionen arbeitet [Emika(2021)], wird die Orientierungsdifferenz $d\vec{r}_Q$ in Euler-Winkel $d\vec{r}_E$ umgerechnet:

$$d\vec{r}_E = 2 \cdot \sin(d\vec{r}_Q) \quad (5.11)$$

Mit Hilfe von zwei PD-Reglern, einen für die Positionssteuerung PD_p und einen für die Rotationssteuerung des Endeffektors PD_r , werden die Soll-Beschleunigungen \vec{a}_p bzw. \vec{a}_r mit den Soll-Geschwindigkeiten \vec{v}_p bzw. \vec{v}_r berechnet. P_p und P_r stellen hierbei das P-Glied für Position bzw. Rotation dar, D_p und D_r entsprechend das D-Glied.

$$\vec{a}_p = P_p \cdot d\vec{p} - D_p \cdot \vec{v}_p \quad (5.12)$$

$$\vec{a}_r = P_r \cdot d\vec{r}_E - D_r \cdot \vec{v}_r \quad (5.13)$$

Die erhaltenen Beschleunigungen werden in einem letzten Schritt zu den kartesischen Soll-Geschwindigkeiten des Endeffektors integriert:

$$\vec{v}_p = \int \vec{a}_p dt \quad (5.14)$$

$$\vec{v}_r = \int \vec{a}_r dt \quad (5.15)$$

Die erhaltenen Geschwindigkeiten \vec{v}_p und \vec{v}_r werden anschließend an die Steuereinheit des Knickarmroboters übertragen.

6 Umsetzung der Automatisierten Funktion, Anomalieerkennung und Lerndatengenerierung

Neben den bereits vorgestellten Modulen für die Teleoperation in Kapitel 5 gibt das folgende Kapitel ein Beispiel für die noch fehlenden Module und demonstriert eine vollständige Umsetzung des in Kapitel 4 vorgestellten Konzepts für das kontinuierliche Lernen von CPS. Hierzu zählt die *Automatisierte Funktion*, die sich aus Objekterkennung, Greifpunktbestimmung und Bestimmung der Ablegeposition zusammensetzt. Hinzu kommt eine *Anomalieerkennung* sowie eine automatisierte *Lerndatengenerierung*, die sowohl während des automatisierten Betriebs als auch bei den Teleoperationsstufen 3 und 4 die notwendigen Daten für eine kontinuierliche Systemverbesserung bereitstellen.

Gemeinsam mit dem vorherigen Kapitel 5 wird die dritte forschungsleitende Fragestellung beantwortet:

Fragestellung 3: Wie lässt sich eine konkrete Umsetzung eines solchen Systems realisieren und welche Module sind dafür notwendig?

6.1 Objekterkennung

Die Objekterkennung basiert auf der *One-Stage One-Shot Object Detection* (kurz OS2D) von Osokin [Osokin et al.(2020)Osokin, Sumin, und Lomakin]. Im Gegensatz zu herkömmlichen Deep-Learning Verfahren in der Objekterkennung [Ren et al.(2016)Ren, He, Girshick, und Sun] [Bochkovskiy et al.(2020)Bochkovskiy, Wang, und Liao], bei denen jedes zu erkennende Objekt einer Klasse zugewiesen ist und eine Vielzahl an annotierten Bildern für ein Training notwendig sind, ist die Grundidee von OS2D eine merkmalbasierte Suche von Referenzen innerhalb einer Szene. Die Struktur von OS2D ist

in Abbildung 6.1 dargestellt und soll an dieser Stelle kurz vorgestellt werden. Für eine detaillierte Beschreibung wird an die Veröffentlichung von Osokin verwiesen [Osokin et al.(2020)Osokin, Sumin, und Lomakin].

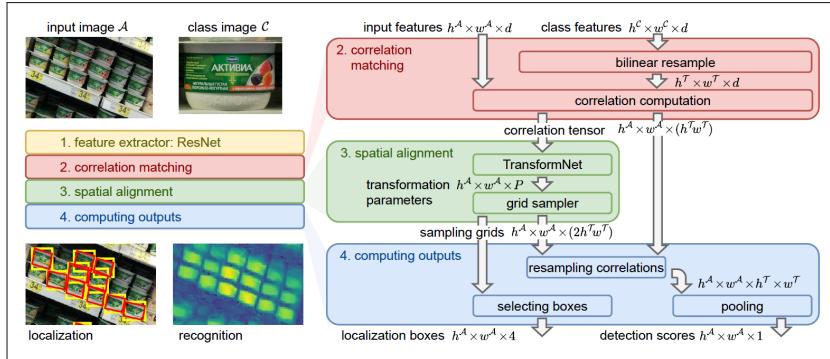


Abbildung 6.1: Struktur der merkmalbasierten Objekterkennung *One-Stage One-Shot Object Detection* (OS2D) [Osokin et al.(2020)Osokin, Sumin, und Lomakin]

Die Merkmalextrahierung findet durch ResNet-50 [He et al.(2015)He, Zhang, Ren, und Sun] statt und führt auf Grund der unterschiedlichen Bildgrößen zwischen Eingangsbild A und Referenzbild C zu zwei unterschiedlichen und zunächst inkompatiblen Merkmalvektoren $\mathcal{A} = h^A \cdot w^A \cdot d$ bzw. $\mathcal{C} = h^C \cdot w^C \cdot d$. Der Merkmalvektor des Referenzbildes C wird aus diesem Grund durch bilineare Interpolation [Jaderberg et al.(2016)Jaderberg, Simonyan, Zisserman, und Kavukcuoglu] auf einen Merkmalvektor mit fester Größe $\mathcal{T} = h^T \cdot w^T \cdot d$ gebracht und der dreidimensionale Korrelationsvektor $c = h^C \cdot w^C \cdot (h^T w^T)$ gebildet.

Anschließend werden affine Transformationen mit Hilfe des von Osokin [Osokin et al.(2020)Osokin, Sumin, und Lomakin] beschriebenen und auf [Rocco et al.(2018)Rocco, Arandjelović, und Sivic] basierenden TransformNets berechnet. In einem letzten Schritt werden die erhaltenen Transformationsparameter und eine quantitative Bewertung der Korrelation zur Bestimmung von Boundingbox und Erkennungsbewertung verwendet.

Für den Einsatz mit einem Kommissionierroboter wurde die von Osokin [Osokin et al.(2020)Osokin, Sumin, und Lomakin] vorgestellte Struktur für die Verwendung mit mehreren Referenzbildern erweitert. Die erhaltenen Korrela-

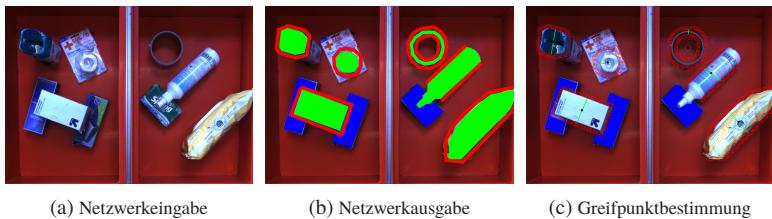


Abbildung 6.2: Rot: mögliche Greifbereiche; Grün: Objekte im Vordergrund; Blau: verdeckte Objekte, eigene Darstellung

tionsvektoren werden entsprechend ihrer quantitativen Bewertung sortiert und die am besten passende Referenz für die Transformation verwendet. In Kapitel 7.3 wird detailliert auf die Verwendung von OS2D und dessen Relevanz für das Lernen aus Teleoperation eingegangen.

6.2 Greifpunktbestimmung

Die verwendete Greifpunktbestimmung basiert auf dem gemeinsam mit Hopfgarten entwickelten Verfahren *Grasp Area Detection of Unknown Objects based on Deep Semantic Segmentation* [Hopfgarten et al.(2020) Hopfgarten, Auberle, und Hein], das im folgenden kurz erläutert werden soll.

Während viele bildbasierte Greifpunktterkennungsverfahren [Rao et al.(2010) Rao, Le, Phoka, Quigley, Sudsang, und Ng] [Lenz et al.(2015) Lenz, Lee, und Saxena] [Kumra und Kanan(2017)] entweder einen 5D-Vektor $\vec{p} = (x, y, \theta, h, w)^T$ für die Beschreibung des Greifpunktes in der Bildebene oder die vollständige 7D Position (3D Position, 3D Orientierung und Öffnungsweite des Greifers) angeben, ist eine genaue Angabe eines *richtigen* Greifpunktes selten möglich. Eine Annotierung von allen guten Greifpositionen, die für ein Training eines Neuronalen Netzes benötigt werden, ist somit kaum möglich. Die meisten Objekte lassen sich mit einer Vielzahl an unterschiedlichen Greifposen robust manipulieren.

Um dieser Problematik entgegenzuwirken, wird anstelle der Berechnung eines oder mehrerer Greifvektoren die Szene in drei unterschiedliche Bereiche aufgeteilt: Objekte im Vordergrund, verdeckte Objekte und potentielle Greif-

	FCN	DeepLab V3+ (mobilenet-v2)	DeepLab V3+ (Xception)
<i>SegRGB</i>	$\bar{t} = 6.1min$	$\bar{t} = 19.93min$	$\bar{t} = 21.3min$
	$\bar{a} = 94.71\%$	$\bar{a} = 94.15\%$	$\bar{a} = 94.51\%$
	$f = 11\,138M$	$f = 3\,277M$	$f = 19\,712M$
<i>SegDepth</i>	$\bar{t} = 6.33min$	$\bar{t} = 16.23min$	$\bar{t} = 22.02min$
	$\bar{a} = 95.65\%$	$\bar{a} = 95.05\%$	$\bar{a} = 96.14\%$
	$f = 11\,080M$	$f = 3\,262M$	$f = 19\,698M$
<i>SegModal</i>	$\bar{t} = 6.83min$	$\bar{t} = 33.73min$	$\bar{t} = 44.72min$
	$\bar{a} = 96.53\%$	$\bar{a} = 95.36\%$	$\bar{a} = 96.51\%$
	$f = 22\,218M$	$f = 6\,539M$	$f = 39\,409M$

Tabelle 6.1: Vergleich von Trainingsdauer \bar{t} , durchschnittliche Genauigkeit \bar{a} und FLOPS pro Inferenz f in Millionen.

flächen von Objekten (Flächen, an denen ein Parallelgreifer einen Kontakt zum Objekt herstellen kann). Anschließend findet eine greiferspezifische Bestimmung der Greifpose in Abhängigkeit des Objektmittelpunkts und Greifbreite statt. Die drei Bereiche sowie die Verarbeitungsschritte sind in Abbildung 6.2 dargestellt.

Es wurden mehrere Verfahren (FCN [Long et al.(2015)Long, Shelhamer, und Darrell], DeepLab V3+ [Chen et al.(2018)Chen, Zhu, Papandreou, Schroff, und Adam] mit mobilenet-v2 und Xception als Backbone) für die Segmentierung untersucht. Zusätzlich wurden entweder nur das Farbbild (*SegRGB*), nur das Tiefenbild (*SegDepth*) oder multimodal mit beiden Eingangsdaten (*SegModal*) trainiert und validiert. Die Trainingsdauer \bar{t} , die Genauigkeit \bar{a} und die Anzahl an FLOPS (engl.: Floating Point Operations Per Second, FLOPS) f sind in Tabelle 6.1 dargestellt.

Es konnte gezeigt werden, dass sich eine Segmentierung mit Hilfe eines kleinen Datensatzes (108 Bilder, max. 11 Objekte je Bild) und den drei vorgestellten Kategorien auch auf bisher ungesehene Objekte übertragen lässt. Wie in Tabelle 6.1 ersichtlich, liefert die Verwendung des multimodalen Ansatzes *SegModal* die besten Ergebnisse. Während FCN [Long et al.(2015)Long, Shelhamer, und

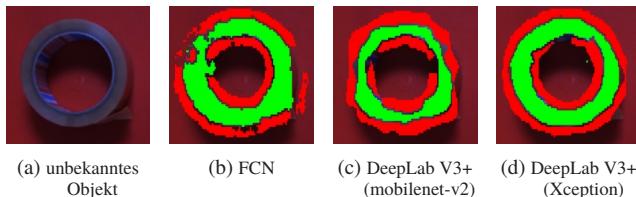


Abbildung 6.3: Vergleich von *SegModal* mit unterschiedlichen Backbones bei einem bisher un-
gesehenen Objekt, eigene Darstellung

Darrell] bei der Genauigkeit etwas besser abschneidet, zeigt Deeplab V3+ (Xception) [Chen et al.(2018)Chen, Zhu, Papandreou, Schroff, und Adam] eine bessere Leistung bei weiteren unbekannten Objekten (Abbildung 6.3). Aus diesem Grund wurde sich an dieser Stelle für die Verwendung des Deeplab V3+ (Xception) Ansatzes in Kombination mit der multimodalen Segmentierung entschieden.

6.3 Ablageposition

Wurde das zu kommissionierende Objekt erkannt und erfolgreich gegriffen, wird eine passende Ablageposition benötigt. Die Berechnung erfolgt mit Hilfe des *MaxRectsBssf* (Maximal Rectangles Best Short Side Fit) Algorithmus [Jylänki(2010)] [secnot(2021)].

MaxRectsBssf berechnet eine platzsparende Positionierung von Rechtecken innerhalb eines größeren, ebenfalls rechteckigen Containers. Soll ein Objekt mit den Abmaßen $R = (w, h)$ in dem Container mit den Abmaßen $F = (W, H)$ abgelegt werden, werden durch die zusätzliche Randbedingung *Best Short Side Fit* neue Objekte stets mit der kürzeren Kante $l = \min(w, h)$ des Objekts an die kürzere Kante des Containers $L = \min(W, H)$ platziert. Anschließend werden die freien Bereiche entlang der Rechteckachsen in neue Subcontainer F_i geteilt, wobei eine Überschneidung ausdrücklich erlaubt ist.

Im Rahmen der Anwendung mit einem Kommissionierroboter können die rechteckigen Boundingboxen der Objekterkennung (siehe Kapitel 6.1) als zu packendes Objekt verwendet werden. Die Abmaße des Containers entspre-

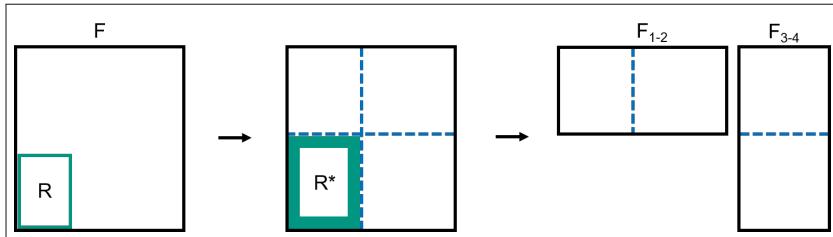


Abbildung 6.4: Funktionalität des *MaxRectsBssf* Algorithmus mit Anpassung für die Verwendung von Parallelgreifern, adaptiert aus [Jylänki(2010)]

chen der verwendeten KLT des Demonstrators (siehe Kapitel 5.1). Auf Grund der Verwendung eines Parallelgreifers wird ein zusätzlicher Abstand der zur platzierenden Objekte sowohl zum Rand der Kiste als auch untereinander hinzugefügt ($R \rightarrow R^*$). Diese Anpassung sowie die beschriebene Aufteilung in Subcontainer ist in Abbildung 6.4 dargestellt.

In einem nächsten Iterationsschritt wird überprüft, ob das neue Objekt in eines der Subcontainer F_i platziert werden kann. Sollte eine Verwendung beider Subcontainer eines Clusters benötigt werden, werden verbundene Cluster entsprechend neu definiert.

6.4 Anomalieerkennung

Das Modul Anomalieerkennung ist für die Sicherstellung eines reibungslosen Ablaufs der gesamten *Automatisierten Funktion* verantwortlich. Hierfür wird ausschließlich auf die Gruppe der regelbasierten Anomalieerkennungen (siehe 3.4) zurückgegriffen.

Folgende Vorgänge werden hierbei untersucht:

1. Objekterkennung
2. Greipunktbestimmung und Greifvorgang
3. Ablegeposition und Ablegevorgang
4. Untersuchung nach beendeter Kommissionierung

Alle Vorgänge werden entsprechend der bereits eingeführten Definition

$$\alpha(p) > \theta \quad (6.1)$$

mit einem geeigneten Schwellwert θ überprüft. Wird eine Anomalie in einer der aufgeführten Vorgänge festgestellt, wird ein Teleoperator zur Lösungsfindung hinzugezogen.

Objekterkennung Handelt es sich bei dem zu erkennenden Objekt um ein bisher unbekanntes, wird direkt die Hilfe eines Teleoperators angefragt. Handelt es sich hingegen um ein bereits bekanntes Objekt mit hinterlegten Referenzbildern (siehe 6.1), wird der Wert der Erkennungsbewertung mit der initialen Erkennungsbewertung des Teach-In Prozesses mit einer Toleranz von 10% untersucht.

Greifpunktbestimmung und Greifvorgang Die Untersuchung der Greifpunkte geschieht durch eine Koordinatentransformation in Roboterkoordinaten mit anschließendem Abgleich des erlaubten Arbeitsraumes. Zusätzlich wird während des Greifvorgangs die Fingerdistanz des Parallelgreifers ausgewertet. Sollte diese nicht > 0 sein, wird von einem Fehler ausgegangen.

Ablegeposition und Ablegevorgang Wie in Kapitel 6.3 beschrieben, werden Objekte in Containern bzw. Subcontainern eingelagert. Sollten die Abmaße eines neuen Objekts die verfügbare Fläche übersteigen oder sich die Ablageposition außerhalb des Arbeitsraumes befinden, wird von einem Fehler ausgegangen.

Untersuchung nach beendeter Kommissionierung Die automatisierte Kommissionierung erfolgt auftragsbasiert. Entsprechend kann nach mutmaßlicher Vollendung der Kommissionierung überprüft werden, ob alle Artikel aus der Quelle entnommen und in der Senke platziert wurden. Dies kann für die Kommissionierung von einzelnen Objekten über die Objekterkennung passieren oder für das vollständige Leeren der Kiste über die Auswertung des Tieffeldbildes. Wird eine Abweichung zum erwarteten Zustand erkannt, wird ein

Teleoperator zur Korrektur durch eine direkte Steuerung (Teleoperationsstufe 3/4) hinzugezogen.

6.5 Lerndatengenerierung

Entsprechend der aktuellen Funktionalität des CPS kann eine Lerndatengenerierung auf drei verschiedenen Ebenen stattfinden:

- Teleoperationsstufe 1 und 2 (direkt eingezeichnet)
- Automatisierte Funktion (erfolgreiche Kommissionierung)
- Zusatzmodul DORI (erfolgreiche Kommissionierung und Teleoperation Stufe 3 und 4)

Teleoperationsstufe 1 und 2 Die Hilfestellung des Teleoperators für die Teleoperationsstufe 1 bzw. 2 findet durch die Verwendung von 2D-Peripherie (siehe Kapitel 4.7) statt. Hierbei kann ein Teleoperator eine Hilfestellung für die Objekterkennung geben, indem eine Boundingbox um das zu greifende Objekt gezeichnet wird. Anschließend findet die automatisierte Bestimmung der Greifpunkte (siehe Kapitel 6.2) statt. Die generierten Lerndaten bestehen somit aus einer Boundingbox auf einem Einzelbild für eine spezifische Objekt-kategorie, die zur Verbesserung der Objekterkennung genutzt werden kann.

Des Weiteren ist ein direktes Einzeichnen der Greifpunkte möglich. Diese können gemeinsam mit der Objekterkennung abgelegt und für eine optimierte Greifpunktbestimmung genutzt werden.

Automatisierte Funktion Die erhaltenen Daten der *Automatisierten Funktion*, also automatisiert generierte Boundingboxen sowie Greifpunkte, können gemeinsam mit dem aufgezeichneten Einzelbild abgelegt werden. Anschließend kann entweder eine Auswahl auf Basis weiterer Kriterien (z.B. Erkennungsbewertung, Greifpunktbewertung, etc.) stattfinden oder die Daten ungefiltert in einen erweiterten Datensatz überführt werden. Dieser beinhaltet erwartungsgemäß keine signifikanten neuen Daten, da die implementierte

Automatisierte Funktion bereits in der Lage war, eine erfolgreiche Kommisionierung durchzuführen. Dennoch ist die Wahrscheinlichkeit sehr hoch, dass bisher im Trainingsdatensatz nicht vorhandene Blickwinkel und Objektpositionen hinzukommen und zu einem robusteren Ergebnis der *Automatisierten Funktion*, wie in Kapitel 7.3 demonstriert, führen.

Zusatzmodul DORI Für die Lerndatengenerierung der Teleoperationsstufen 3 und 4, die eine direkte Steuerung des Roboters mit sich bringen, ist die Verwendung eines zusätzlichen Moduls zur Lerndatengenerierung notwendig. Das verwendete Verfahren *Dual-shot object re-identification from RGB-D images (DORI)* wurde durch eine Masterarbeit von Knievel [Knievel(2020)] im Rahmen dieser Dissertation entwickelt, anschließend gemeinsam in das erarbeitete Framework integriert und als Konferenzbeitrag eingereicht [7]. Eine Übersicht über die Funktionalität von DORI soll im folgenden vorgestellt werden, wobei für eine detaillierte Beschreibung der Implementierung an das eingereichte Paper verwiesen wird.

Abbildung 6.5 zeigt die entwickelte Struktur, die in großen Teilen aus Modulen des Mask R-CNN Frameworks [He et al.(2018)He, Gkioxari, Dollár, und Girshick] besteht. Während Mask R-CNN insbesondere für Instanzsegmentierung von bekannten Objekten entwickelt wurde, soll mit DORI ein Klassifikator entwickelt werden, der unbekannte Objekte aus zwei Eingangsbildern (vor und nach der Kommissionierung) erkennt und anhand der erkannten Bewegung segmentiert. Hierbei ist es möglich, dass sich neben dem kommissionierten Objekt weitere Objekte in der Szene bewegen. Ein einfacher Abgleich zwischen Vorher und Nachher würde somit nicht das gewünschte Ergebnis liefern.

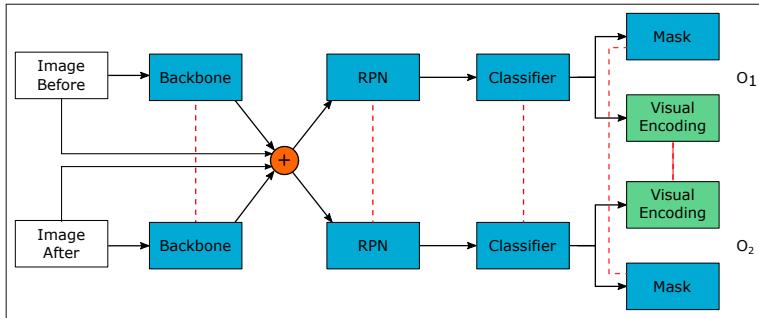


Abbildung 6.5: Überblick über die Netzwerkarchitektur von DORI: Blaue Module sind von Mask R-CNN adaptiert, grüne Module wurden hinzugefügt. Rot gestrichelte Linien zeigen Gewichte, die zwischen den Stufen geteilt werden. Es sind verschiedene Möglichkeiten zur Kombination von Bild- und Merkmalinformationen denkbar (Knotenpunkt in Orange) [7]

Die Merkmalextrahierung der beiden Eingangsbilder I_1 und I_2 erfolgt in dem auf ResNet35 [He et al.(2015)He, Zhang, Ren, und Sun] und FPN (Feature Pyramid Network) [Lin et al.(2017)Lin, Dollár, Girshick, He, Hariharan, und Belongie] basierenden Backbone für das Farb- und Tiefenbild der verbauten Teleoperationskamera (vgl. Kapitel 5.1). Hierbei erfolgt eine Konkatenation der erhaltenen Merkmalpyramiden von Farb- und Tiefenbild. Anschließend findet eine Konkatenation des Merkmalsvektors F_1 von I_1 mit einer erstellten Bildpyramide der Bilddifferenz $I_1 - I_2$ statt. Der erhaltene Merkmalvektor wird anschließend durch das *Region Proposal Network* (RPN) sowie einen binären Klassifikator (bewegtes Objekt / Hintergrund) weiterverarbeitet und maskiert. Das erhaltene Set O_1 beinhaltet somit eine Boundingbox sowie eine Maskierung aller bewegten Objekte für I_1 .

Für I_2 werden alle beschriebenen Schritte entsprechend umgekehrt durchgeführt und man erhält ebenfalls eine Boundingbox sowie eine Maskierung aller bewegten Objekte für I_2 . Die Zuordnung der bewegten Objekte O_1 und O_2 wird durch eine gewichtete Kombination $d_{n,m}$

$$d_{n,m} = (1 - a) \cdot v_{n,m} + a \cdot p_{n,m} \quad (6.2)$$

bestimmt. Die Kombination aus visueller Distanz $v_{n,m}$ und physischer Distanz der Greiferpositionen $p_{n,m}$ erlaubt eine robuste Zuweisung bewegter Objekte

aus den Objektpaaren O_1 und O_2 . Da im vorliegenden Fall keine Objekte übereinander gelegt werden, wurde eine doppelten Gewichtung der physischen Distanz der Greiferposition $p_{n,m}$ gewählt. Eine detaillierte Beschreibung der visuellen und physischen Distanz findet sich in [7].

Abbildung 6.6 zeigt die erfolgreiche Zuordnung von Objekt 2, trotz starker optischer Veränderung durch die Kommissionierung. Zusätzlich wurde eine Bewegung von Objekt 1 festgestellt, das durch die Kommissionierung verschoben wurde. Neben der Verwendung von DORI für die Lerndatengenerierung bei den Teleoperationsstufen 3 und 4 kann DORI auch während des *automatisierten Betriebs* die Objekterkennung (siehe Kapitel 6.1) durch automatisiert erstellte Lerndaten kontinuierlich verbessern.



Abbildung 6.6: Kommissionierte Objekte werden in beiden Bildern erkannt und segmentiert (links: vorher, rechts: nachher) [7]

7 Versuchsdurchführung, Funktionsdemonstration und Bewertung

Das Kapitel *Versuchsdurchführung, Funktionsdemonstration und Bewertung* gliedert sich in drei unterschiedliche Versuche, die jeweils in einem Unterkapitel behandelt werden:

1. Systemlatenz
2. Studie über den Einfluss von VR und Latenz
3. Versuchsaufbau Lernen durch Teleoperation

Das erste Unterkapitel *Systemlatenz* gibt Auskunft über die internen Parameter des in Kapitel 5 vorgestellten Systems zur Teleoperation. Die Frage, mit welchen Latzenzen Steuersignale von Teleoperator zu CPS übertragen und umgesetzt werden, wird hier beantwortet.

Aufbauend auf den Ergebnissen des Versuchs zur *Systemlatenz*, wird im zweiten Unterkapitel *Studie über den Einfluss von VR und Latenz* eine Teleoperation vor Ort ohne zusätzliche Latzenzen mit einer (entkoppelten) Steuerung in VR mit mehreren, künstlich vorgegebenen, Latzenzen verglichen. Hierfür werden durch eine Proband*innenstudie Fragen über die Intuitivität der Steuerung, den Einfluss einer Visualisierung und Steuerung in VR untersucht und die Forschungsbegleitende *Fragestellung 4: Kann eine Hilfestellung des Menschen via Teleoperation bei komplexen Aufgabenstellungen benutzerfreundlich und intuitiv gestaltet werden?* beantwortet.

Weiterhin wird auf Basis der durchgeführten Proband*innenstudie eine Antwort auf die Forschungsbegleitende *Fragestellung 5: Wie ist der Einfluss von Latenz zwischen Teleoperator und CPS?* gegeben.

Im dritten und letzten Unterkapitel *Lernen durch Teleoperation* wird, auf Basis des in Kapitel 4 vorgestellten Konzepts, die Lernfähigkeit des Gesamtsystems untersucht. Mit einem dritten Versuch wird die Forschungsbegleitende **Fragestellung 6: Kann durch das entwickelte Framework die Leistungsfähigkeit eines CPS mittelbar und unmittelbar verbessert werden?** beantwortet.

7.1 Systemlatenz

Die Systemlatenz L beschreibt die Gesamlatenz zwischen Sollposition auf Seite des Teleoperators und Ist-Position auf Seite des CPS. Sie setzt sich aus folgenden Komponenten zusammen: Der Controller-Latenz L_C zwischen VR-Controller und PC des Teleoperators, der Netzwerklatenz L_N zwischen PC des Teleoperators und dem Steuer-PC des CPS und der Latenz des Roboterarms L_R , zwischen Sollposition am Steuer-PC und realer Position des Roboterarms.

Die tatsächliche Latenz zwischen der Position des VR-Controllers (Eingabedaten des Teleoperators, siehe 5.2.2) und der Endeffektorposition des zu steuernden Roboters wird durch Verwendung eines zweiten Knickarmroboters möglich. Am Endeffektor des zweiten Roboters wird ein VR-Controller montiert, der eine Teleoperation simuliert. Durch die Verwendung des zweiten Roboters ist eine exakte Position des VR-Controllers inklusive Zeitstempel vorhanden. Der Aufbau ist in Abbildung 7.1 dargestellt.

Die Übertragungszeit der Funkverbindung zwischen VR-Controller und PC des Teleoperators ist eine unbekannte Größe, die nicht direkt gemessen werden kann. Hierzu fehlen exakte Zeitstempel sowie eine genaue Referenzposition des VR-Controllers (Abbildung 7.2). Durch das Hinzuziehen eines zweiten Roboters und die Befestigung des VR-Controllers am Endeffektor, lassen sich die Latenz des VR-Controllers L_C und die Latenz der Netzwerkübertragung L_C jedoch kombiniert messen. Die Verzögerung der Endeffektorposition zwischen beiden Robotern beinhaltet die vollständige Systemlatenz L . Hierfür werden beide Roboter vom Steuer-PC des CPS aus angesteuert, um eine vergleichbare Zeitsynchronisierung der Stellbefehle an den zweiten Roboter mit den ausgelesene Endeffektorpositionen des teleoperierten Roboters zu erhalten.



Abbildung 7.1: Latenzmessung durch VR-Controller an einem zweiten Knickarmroboter, eigene Darstellung

Die Berechnung der Systemlatenz L

$$L = L_C + L_N + L_R \quad (7.1)$$

vereinfacht sich mit

$$L_N^* = L_C + L_N \quad (7.2)$$

zu

$$L = L_N^* + L_R \quad (7.3)$$

Der schematische Aufbau der Versuchsdurchführung ist in Abbildung 7.2 dargestellt.

Während die Endeffektorposition (EE) von Roboter 1 als Referenzposition direkt an den Steuer-PC des CPS gesendet wird, müssen die aufgenommenen Daten des VR-Controllers zunächst an den Steuer-PC des Teleoperators und anschließend via Netzwerk an den Steuer-PC des CPS übertragen werden. Die

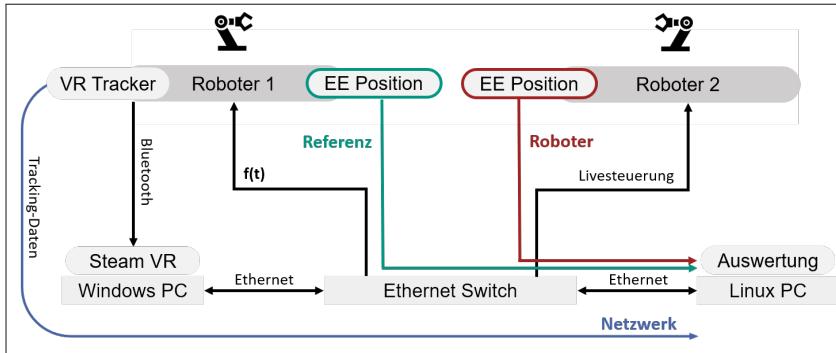


Abbildung 7.2: Schematische Darstellung von Versuchsaufbau 1 mit zwei Knickarmrobotern, dem Referenzsignal (grün), der Netzwerkverzögerung inklusive Verzögerung des VR-Controllers (blau) sowie der tatsächlichen Position des teleoperierten Roboters (rot) und dem Stellsignal $f(t)$ an Roboter 1, eigene Darstellung

erhaltenen Daten werden dort via der in Kapitel 5.2.6 vorgestellten Robotersteuerung an Roboter 2 übermittelt.

Hieraus lassen sich drei Signale ableiten: Die Referenzposition (im folgenden als *Referenz* bezeichnet), die Position des VR-Controllers L_N^* inklusive der Latenz des VR-Controllers L_C und der Latenz der Netzwerkuübertragung L_N (im folgenden als *Netzwerk* bezeichnet) sowie die Verzögerung zwischen Steuer-PC und Endeffektorposition (EE) des teleoperierten Roboter 2 (im folgenden als *Roboter* bezeichnet). Die Latenz zwischen beiden Endeffektorpositionen ist die gesuchte Systemlatenz L . Die Bewegungen von Roboter 1 werden durch ein Testsignal $f(t)$ vom Steuer-PC des CPS gesteuert.

Das Testsignal $f(t)$ soll die Bewegungen eines Teleoperators simulieren und wurde auf Basis der gemessenen Bewegung bei einer realen Teleoperation ausgewählt (Abbildung 7.3). Die ermittelte maximale Geschwindigkeit des Roboters beträgt hierbei $v_{max} = 0.6265m/s$ und die maximale Beschleunigung $a_{max} = 19.95m/s^2$.

Eine Frequenzanalyse in Abbildung 7.3 ergibt, dass insbesondere Frequenzen bis 0.5 Hz bei einer teleoperierten Artikelkommissionierung relevant sind. Mit dem Testsignal $f(t)$ und den Periodendauern $p = [5, 10, 15, 20]$ werden sowohl Steuerbefehle mit hohen Geschwindigkeiten und einer niedrigen Fre-

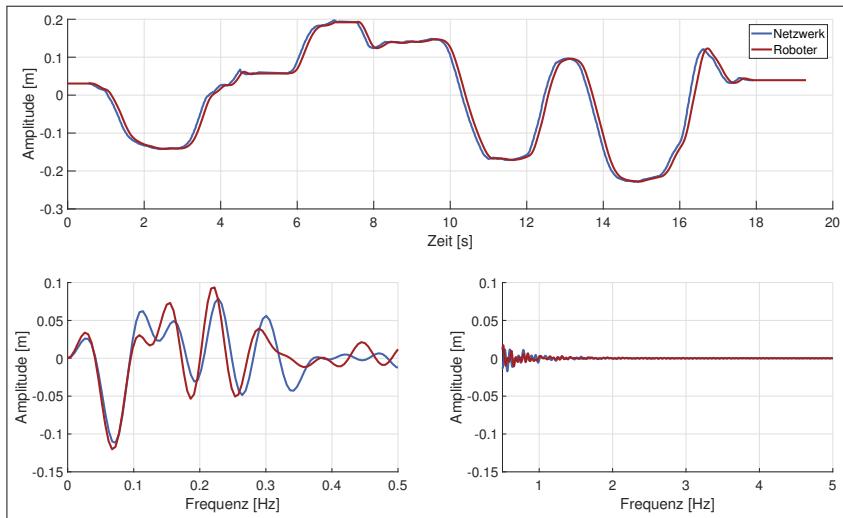


Abbildung 7.3: Veranschaulichung der Roboterbewegung in Relation zu den Eingangsdaten des Tracking-Devices bei einer teleoperierten Artikelkommissionierung. $v_{max} = 0.62 \text{ m/s}$, $a_{max} = 19.95 \text{ m/s}^2$. Die relevanten Frequenzen befinden sich zwischen 0.0 Hz und 0.5 Hz, eigene Darstellung

quenz als auch Steuerbefehle mit höheren Frequenzen und niedrigeren Maximalgeschwindigkeiten simuliert.

$$f(t) = 0.3 \cdot \sin(\pi \cdot (1 - \cos(\frac{\pi}{p} \cdot t))) \quad (7.4)$$

Das Signal ist in Abbildung 7.4 beispielhaft mit der Periodendauer $p = 10$ dargestellt.

Der Versuch wurde mit jeweils drei Wiederholungen mit allen vier Periodendauern p durchgeführt. Das Ergebnis der Latenzuntersuchung ergab eine Verzögerung zwischen Referenzposition des Endeffektors von Roboter 1 und der Position des teleoperierten Roboters 2 von 100 ms. Eine Spektralanalyse mit dem Testsignal $f(t)$ und den vier Periodendauern $p = [5, 10, 15, 20]$ zeigte eine nahezu perfekte Übereinstimmung zwischen Referenzsignal von Roboter 1 und gemessener Bewegung des Endeffektors von Roboter 2 (Abbildung 7.5).

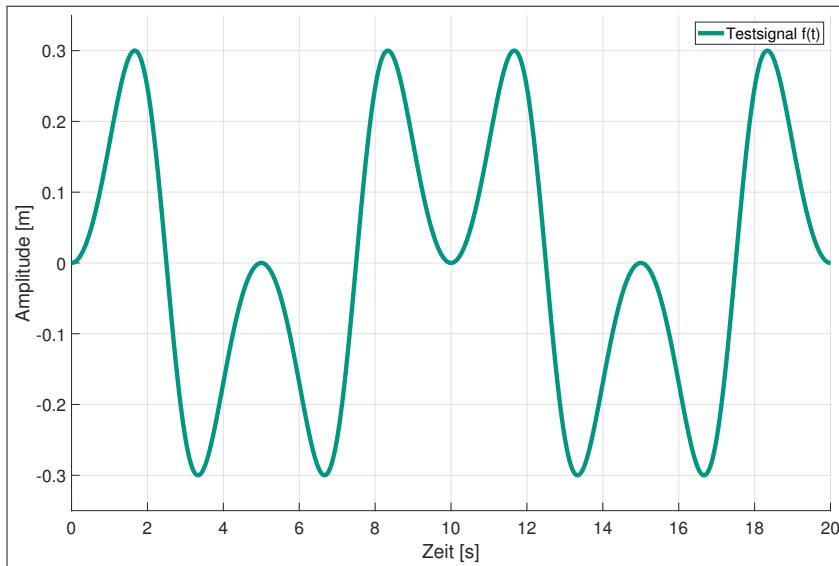


Abbildung 7.4: Darstellung des Testsignals $f(t)$ mit Periodendauer $p = 10$, eigene Darstellung

An dieser Stelle lässt sich die Aussage treffen, dass sowohl das Eingangssignal des VR-Controllers als auch die in Kapitel 5.2.6 vorgestellte Robotersteuerung erwartungsgemäß funktionieren und für die Teleoperation eines Kommissi-
nierroboters geeignet sind.

Eine weitere Versuchsreihe soll nun darüber Aufschluss geben, ob die gemessene Verzögerung von 100 ms sich linear zu weiteren, künstlichen Verzögerungen verhält. Hierfür wurde erneut auf das Testsignal $f(t)$ mit einer Periodendauer von $p = 10$ zurückgegriffen und die Netzwerkverzögerung L_N künstlich um 0 ms, 50 ms, 100 ms, 250 ms, 500 ms und 1000 ms erhöht. Ein Beispiel der gemessenen Signale ist in Abbildung 7.6 dargestellt.

Das Ergebnis ist eindeutig: Die künstliche Latenz verhält sich erwartungsgemäß linear zu der bereits gemessenen Systemlatenz von 100 ms (Abbildung 7.7). Für alle weiteren Versuche wird somit eine minimale Latenz von 100 ms angenommen (keine zusätzliche künstliche Latenz) und jede weitere, künstliche Latenz addiert. Die gewonnenen Erkenntnisse über die Systemlatenz

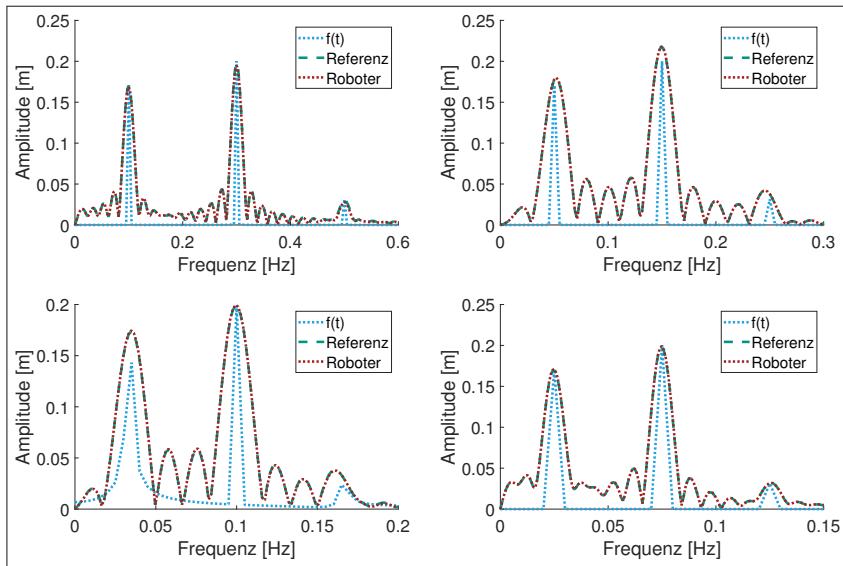


Abbildung 7.5: Frequenzanalyse zwischen Roboter und Referenzsignal, eigene Darstellung

dienen als Grundlage für die *Probandenstudie über den Einfluss von VR und Latenz* in Kapitel 7.2.

7.2 Proband*innenstudie über den Einfluss von VR und Latenz

Nachdem die Systemlatenz (vgl. Abschnitt 7.1), also die Verzögerung zwischen Eingabedaten des VR-Controllers und Endeffektor des zu teleoperierenden Roboters, auf 100 ms bestimmt wurde, erfolgte die Durchführung einer Proband*innenstudie mit 42 Teilnehmer*innen und insgesamt 1176 erfolgreichen Teleoperationen. Die Studie soll Aufschluss über die Intuitivität der Steuerung und insbesondere über den Einfluss der Latenz während einer Teleoperation der Stufe 4 geben. Die Steuerung wurde mit der in Abbildung 5.7 vorgestellten Anordnung getestet.

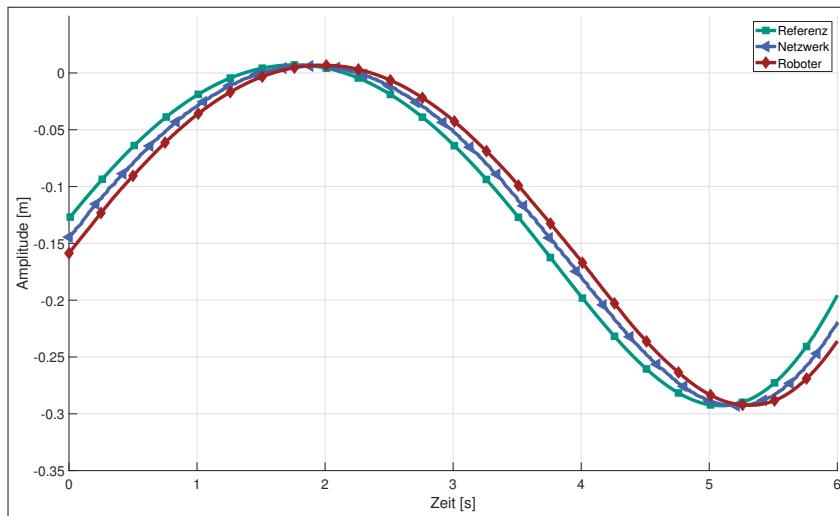


Abbildung 7.6: Gemessene Latenzen bei einer künstlichen Netzwerkverzögerung von 100 ms mit dem Testsignal $f(t)$ und $p = 10$, eigene Darstellung

Die Proband*innen haben zu Beginn eine kurze Einweisung in das System erhalten und durften den Roboter für ca. 30 Sekunden teleoperieren, bevor mit der Studie begonnen wurde. Das Greifen eines Objektes wurde jedoch untersagt, lediglich das Verständnis der Steuerung wurde sichergestellt. Des Weiteren wurden die Proband*innen nach dem *Alter*, dem *Geschlecht*, dem aktuellen *Berufsfeld* (entweder Büro oder Handwerk) gefragt. Eine Selbsteinschätzung auf einer Skala von 0 bis 4 (0 = überhaupt keine Kenntnisse bis 4 = fundierte Kenntnisse) wurde in folgenden Fragestellungen ebenfalls vorgenommen:

1. Erfahrung im Umgang / mit der Programmierung von Robotern [0 bis 4]
2. Erfahrung mit 3D-Anwendungen am Computer, z.B. CAD oder Videospiele [0 bis 4]
3. Erfahrung mit VR-Anwendungen [0 bis 4]
4. Wurde dieser Demonstrator bereits teleoperiert [0 bis 4]

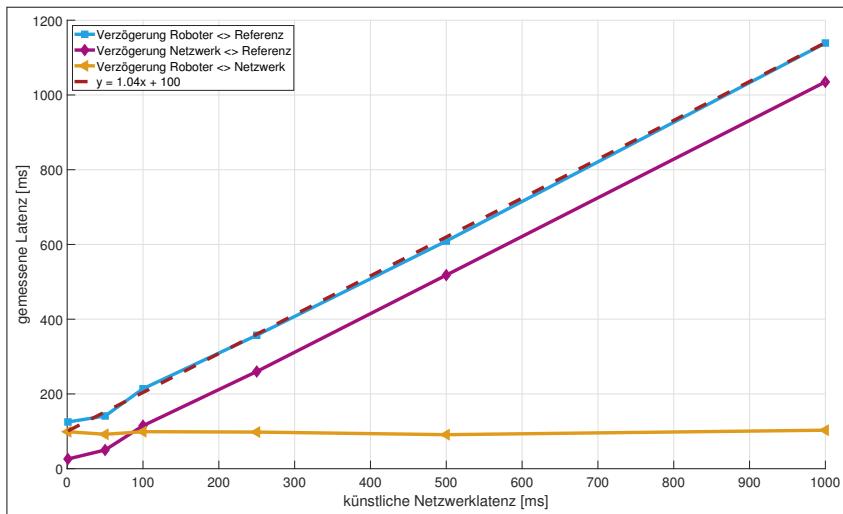


Abbildung 7.7: Linearer Zusammenhang zwischen Systemlatenz und künstlich erhöhte Netzwerklatenz, eigene Darstellung

Eine Übersicht des Datensatzes ist in Tabelle 7.1 dargestellt. Die Erfahrungsbewertung wird in Prozent mit Bezug zur maximalen Punktzahl (4) angegeben.

Die zu erledigende Aufgabe (Abbildung 7.8) ist die Kommissionierung von zwei Würfeln und simuliert eine Kommissionierung mit zwei unterschiedlichen Ablagetoleranzen. Die Aufgabe gliedert sich in zwei Unteraufgaben (erster Würfel mit hoher Ablagetoleranz, zweiter Würfel mit geringer Toleranz), die von den Proband*innen in beliebiger Reihenfolge erledigt werden konnte. Beide Unteraufgaben werden nicht gesondert betrachtet, da eine Kommissionierung von mehreren Objekten (zwei) ohne Unterbrechung untersucht werden soll. Alle angegebenen Versuchszeiten beziehen sich auf die vollständige Durchführung beider Aufgaben. Versuche, bei denen eine Kollision des Roboters mit der Umgebung vorgekommen ist, wurden als Fehlversuche gezählt und der Versuch wiederholt.

Die Gesamtzahl der Teleoperationen im Datensatz bezieht sich auf die Summe aller erfolgreichen sowie nicht erfolgreichen Teleoperationen. Die Auswertung

Anzahl Proband*innen	42
Durchgeführte Teleoperationen	1258
davon erfolgreiche Teleoperationen	1176
Anzahl Fehlversuche	82
Durchschnittsalter	30.47 Jahre
Anteil männlich	64.3 % (27)
Anteil weiblich	35.7 % (15)
Berufsfeld Büro	92.8 % (39)
Berufsfeld Handwerk	7.2 % (3)
<hr/>	
Erfahrung Robotik (1)	33.93 %
Erfahrung 3D-Anwendungen (2)	54.76 %
Erfahrung VR-Anwendungen (3)	36.31 %
Erfahrung Teleoperation (4)	24.4 %

Tabelle 7.1: Tabellarische Übersicht über den erhobenen Datensatz bei der Untersuchung des Einflusses von Latenz auf die Teleoperation der Stufe 4 eines Kommissionierroboters. Die Erfahrungsbewertung wird in Prozent mit Bezug zur maximalen Punktzahl der Umfrage angegeben.

bezieht sich jedoch lediglich auf die erfolgreichen Versuche, mit Ausnahme der Angabe der Anzahl an Fehlversuchen.

Die erste Teilaufgabe ist das teleoperierte Greifen eines gelbfarbigen Würfels mit einer Kantenlänge von 4.2 cm und einer Diagonale von 5.94 cm . Die Ablage des gelbfarbigen Würfels ist ca. 30 cm entfernt, ebenfalls quadratisch und besitzt eine Diagonale von 12 cm . Die Ablagefläche für den gelben Würfel entspricht somit exakt der vierfachen Fläche des Würfels. Die Teilaufgabe mit dem gelben Würfels kann deshalb als die leichtere Aufgabe bezeichnet werden.

Die zweite Aufgabe mit einem blauen Würfel gleicht der ersten, bis auf die Erhöhung der Distanz zwischen Aufnahme- und Ablagepunkt auf 42 cm und

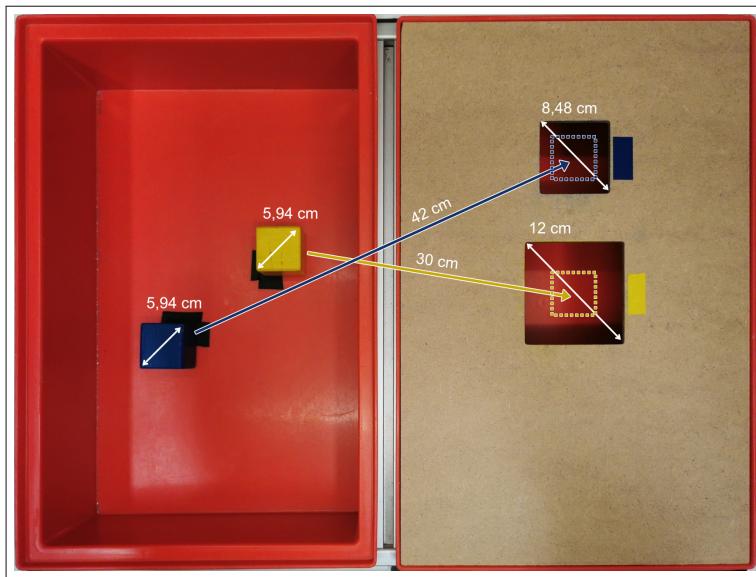


Abbildung 7.8: Aufgabe der Proband*innen, die durch Teleoperation (Stufe 4) durchzuführen ist, eigene Darstellung

die Verringerung der Diagonale der blauen Ablagefläche auf 8.48 cm . Die Ablagefläche des blauen Würfels entspricht somit lediglich der doppelten Fläche. Durch die höhere Distanz zwischen Aufnahme und Ablagepunkt sowie einer niedrigeren Ablagetoleranz kann die Aufgabe mit dem blauen Würfel als die schwierigere Aufgabe bezeichnet werden.

Die Aufgabe wurde von den Proband*innen in 10 unterschiedlichen Varianten durchgeführt. Der erste Versuch wurde ohne VR-Brille und ohne zusätzliche künstliche Verzögerung durchgeführt und pro Proband*in sechsmal wiederholt. Im zweiten Versuch konnten die Probanden die Umgebung lediglich durch die Visualisierung in der virtuellen Realität betrachten und der Versuch wurde ebenfalls sechsmal durchgeführt. Bei allen weiteren Versuchen wurde zusätzlich eine künstliche Verzögerung auf die Steuersignale des Teleoperators sowie auf die Umgebungsvisualisierung gelegt.

Die Proband*innen hatten sowohl bei den Steuersignalen als auch bei der visuellen Rückmeldung über Bewegungen bzw. mögliche Veränderungen der Umgebung eine Verzögerung. Das Systemverhalten entspricht somit einer realistischen Netzwerklatenz zwischen Teleoperator und CPS. Technisch wurde dies durch die in Kapitel 5.2.4 vorgestellten Ringbuffer realisiert, deren Puffergröße entsprechend der gewünschten künstlichen Latenz angepasst wurde.

Versuchsnummer	Verzögerung	Visualisierung	Anzahl Iterationen
V1	100* ms	Realität	6
V2	100* ms	virtuelle Realität	6
V3 - V10	150* - 600* ms	virtuelle Realität	2

Tabelle 7.2: Beschreibung der Visualisierung und vorgegebenen Latenz bei Versuch 1 bis Versuch 10. (*): Die Verzögerung basiert auf der ermittelten Systemlatenz in Kombination mit einer zusätzlichen künstlich vorgegebenen Netzwerklatenz bei den Versuchen V3-V10.

Die mittleren Versuchszeiten aller durchgeführten Versuche sind in Tabelle 7.3 aufgeführt, eine graphische Darstellung der durchschnittlichen minimalen Versuchszeit und die zugehörige Standardabweichung ist in Abbildung 7.9 dargestellt.

Versuch	Latenz[ms]	1	2	3	4	5	6	avg	std	Steigerung[%]	Fehlversuche[%]
V1	100	31.2	23.0	20.5	17.3	16.3	15.4	20.6	10.9	-	3.2 %
V2	100	25.4	24.0	21.9	19.3	20.7	18.7	21.7	9.6	5.3 %	7.9 %
V3	150	22.2	22.5	-	-	-	-	22.3	10.2	3.2 %	5.9 %
V4	200	26.1	25.8	-	-	-	-	26.0	12.9	16.1 %	5.9 %
V5	250	31.7	28.1	-	-	-	-	29.9	16.5	15.0 %	7.1 %
V6	300	33.3	32.6	-	-	-	-	32.9	15.0	10.4 %	7.1 %
V7	350	32.9	33.5	-	-	-	-	33.2	13.0	0.6 %	5.9 %
V8	400	37.3	37.0	-	-	-	-	37.2	15.3	12.0 %	14.3 %
V9	500	41.7	41.6	-	-	-	-	41.6	16.4	12.1 %	9.5 %
V10	600	47.2	49.8	-	-	-	-	48.5	18.8	16.3 %	8.3 %

Tabelle 7.3: Ermittelte Teleoperationszeiten aller Proband*innen und Versuche. Versuch 1 und Versuch 2 wurden mit jeweils sechs Iterationen durchgeführt, Versuch 3 bis Versuch 10 mit jeweils zwei Iterationen. Durchschnitt (avg) und Standardabweichung (std) beziehen sich auf alle durchgeführten Iterationen je Versuch. Die Steigerungen sind prozentual zum vorherigen Versuch angegeben. Die Fehlversuche sind prozentual zur Anzahl an durchgeführten Teleoperationen je Versuch angegeben. Alle Latzen sind in Millisekunden [ms] und alle Versuchszeiten sind in Sekunden [s] angegeben.

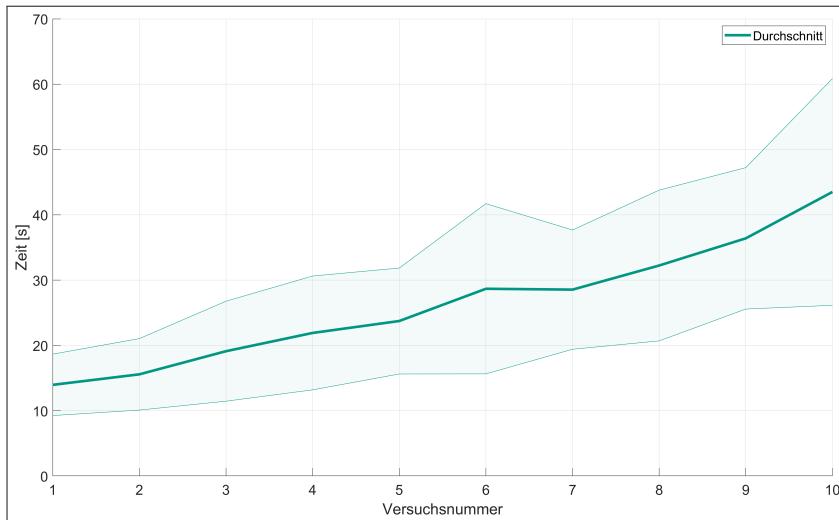


Abbildung 7.9: Einfluss von Latenz und Visualisierung in VR auf die benötigte Teleoperationszeit.
Basierend auf der durchschnittlichen, minimalen Versuchszeit mit zugehöriger Standardabweichung aller Proband*innen und Versuche, eigene Darstellung

Bevor eine detaillierte Auswertung der einzelnen Versuchszeiten vorgenommen wird, lässt sich auf Basis der erhobenen Daten die Aussage treffen, dass eine Teleoperation der Stufe 4 von allen Proband*innen in kürzester Zeit verstanden und ausgeführt werden konnte.

Lediglich 14 der 42 Proband*innen geben einen Wert größer als 1 auf die Frage der Erfahrung im Bereich Robotik an. Im Rückschluss können 28 (66.6 %) der Proband*innen als absolut unerfahren im Umgang, der Bedienung und Programmierung von CPS eingestuft werden. Ebenfalls geben lediglich 14 Proband*innen an, bereits Erfahrung mit VR-Anwendungen zu haben. Der Großteil verwendet somit zum ersten mal eine VR-Brille in Kombination mit einem VR-Controller und wird direkt mit der Teleoperation eines Knickarm-roboters beauftragt.

Von den 14 erfahrenen Proband*innen geben 9 an, den entwickelten Demonstrator bereits in einem früheren Stadium (ohne die Visualisierung mit VR-Brille) teleoperiert zu haben. Der vorliegende Datensatz beinhaltet somit im Bereich Robotik und Teleoperation überwiegend unerfahrene Proband*innen,

die in ihrem Alltag nicht bis selten mit CPS in Kontakt kommen. Der dennoch nicht unerhebliche Anteil an Proband*innen, die bereits eine Teleoperation durchgeführt haben (9 von 42, bzw. 21 %) ist keinesfalls als negativ zu betrachten, sondern bietet möglicherweise eine bessere Aussage auf den Einfluss der Latenz als bei unerfahrenen Benutzern, da die Lernkurve der erfahreneren Proband*innen eher flacher verläuft.

Die Antwort auf die **Forschungsbegleitende Fragestellung 4: Kann eine Hilfestellung des Menschen via Teleoperation bei komplexen Aufgabenstellungen benutzerfreundlich und intuitiv gestaltet werden?** lässt sich, für die Teleoperation eines Kommissionierroboters, bei der gestellten Aufgabe und des erhobenen Datensatzes, ohne Zweifel mit *Ja* beantworten.

Von der Auswertung der Versuchszeiten lassen sich vier Erkenntnisse ableiten, die im Folgenden genauer erörtert werden:

- Qualitativer Einfluss von Latenz auf die Teleoperationsdauer (7.2.1)
- Eine Visualisierung in VR hat Einfluss auf die Teleoperationsdauer (7.2.2)
- Lernfähigkeit der Proband*innen (7.2.3)
- Quantitative Bewertung und Modellierung des Zusammenhangs von Teleoperationsdauer und Latenz für Pick-and-Place Aufgaben (7.2.4)

7.2.1 Qualitativer Einfluss von Latenz auf die Teleoperationsdauer

Die Aussage *Je höher die Latenz, desto höher die Teleoperationsdauer* ist keinesfalls überraschend und kann im Rahmen der durchgeföhrten Studie als belegte Aussage bezeichnet werden.

Qualitativ betrachtet lassen sich deutliche Unterschiede bei gefahrener Trajektorie und Endeffektorgeschwindigkeit des Knickarmroboters in Abhängigkeit von der Latenz beobachten (Abbildung 7.10).

Der Effekt der Übersteuerung des Knickarmroboters zeigt sich bereits im Vergleich von V1 und V2 (7.10a und b), also ohne künstliche Latenz, aber

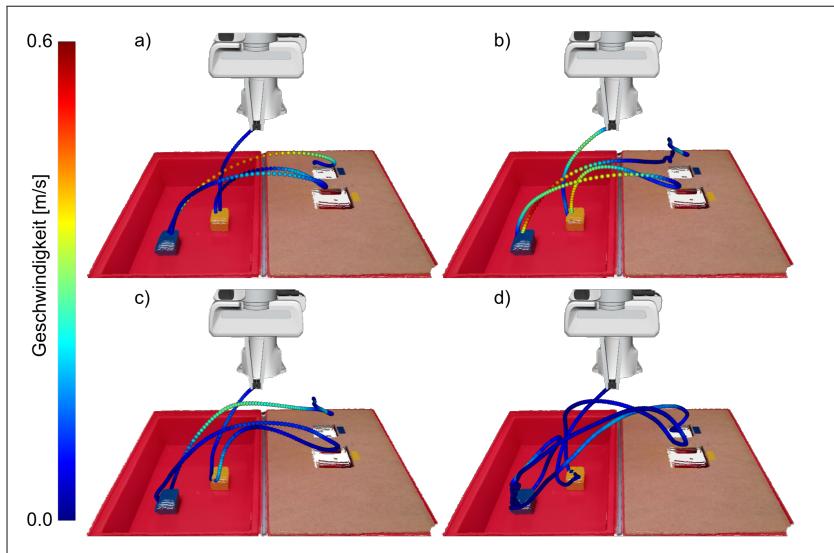


Abbildung 7.10: Darstellung des schnellsten Versuchs mit Robotergeschwindigkeit, teleoperierter Trajektorie und Versuchsdauer in Abhängigkeit von Visualisierung und Latenz bei einer Teleoperation der Stufe 4, eigene Darstellung

- a) ohne VR-Brille, 100 ms Verzögerung, Dauer: 5.2 s
- b) mit VR-Brille, 100 ms Verzögerung, Dauer: 5.8 s
- c) mit VR-Brille, 250 ms Verzögerung, Dauer: 8.1 s
- d) mit VR-Brille, 600 ms Verzögerung, Dauer: 29.4 s

einmal ohne VR-Brille und einmal mit VR-Brille. Der Effekt wird genauer in Abschnitt 7.2.2 untersucht.

Mit steigender Latenz (7.10c) verstärkt sich die Übersteuerung weiterhin. Ab einer Latenz von 600 ms ist eine saubere Ansteuerung von Aufnahme- und Ablageposition nicht mehr gegeben (7.10d). Die Geschwindigkeit wurde durch den Teleoperator deutlich reduziert, dennoch ist eine Übersteuerung und eine ständige Korrektur der Bewegung für eine erfolgreiche Durchführung der Aufgabe notwendig.

Weiterhin stellt sich die Frage, welche Proband*innen niedrige und welche hohe Versuchszeiten erzielen konnten. Hierfür wurde die Gesamtversuchsdauer t_G , also die Summe aller Versuchszeiten einer* Proband*in

$$t_G = \sum_{i=1}^{10} t_{V_i} \quad (7.5)$$

mit den erhobenen Metadaten (siehe Tabelle 7.1) verglichen. Eine Auswertung auf Basis des Berufsfelds ist auf Grund der geringen Stichprobengröße nicht durchgeführt worden. Die Sensitivitätsanalyse auf Basis des *Generalisierten Linearen Regressionsmodells* [Nelder und Wedderburn(1972)] ergibt 3 relevante Kriterien (absteigend sortiert nach Relevanz):

- Erfahrung mit 3D-Anwendungen
- Erfahrung mit VR-Anwendungen
- Erfahrung im Umgang mit Robotern

Der Einfluss dieser Kriterien auf die Gesamtversuchsdauer t_G ist in Abbildung 7.11 dargestellt. Das bedeutet, dass das Geschlecht der Proband*innen und bereits gemachte Erfahrungen mit dem Demonstrator bzw. der Teleoperation eine untergeordnete Rolle spielen. Besonders relevant ist die Erfahrung mit 3D-Anwendungen, die Erfahrung mit VR-Anwendungen und die Erfahrung im Umgang mit Robotern.

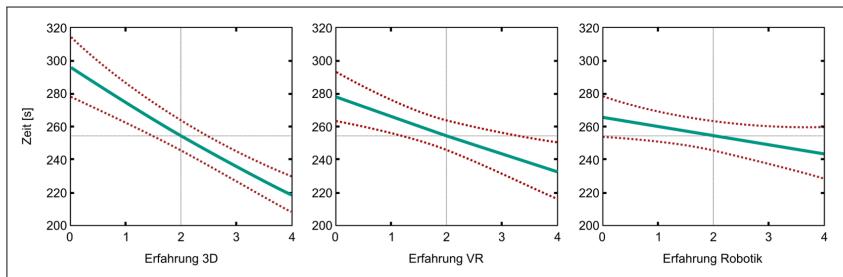


Abbildung 7.11: Relevante Einflussfaktoren auf die Teleoperationszeit der Proband*innen. Je steiler die grüne Kurve, desto höher der Einfluss, eigene Darstellung

Während die ersten beiden Punkte (Erfahrung mit 3D-Anwendungen und Erfahrung mit VR-Anwendungen) in Anbetracht der zu lösenden Aufgabe eher

selbsterklärend erscheinen, ist es überraschend, dass sich eine Erfahrung im Umgang mit Robotern positiv auf die Gesamtversuchsdauer t_G auswirkt.

Eine Hypothese ist die Überschneidung der Proband*innen mit Erfahrung im Umgang mit Robotern und Erfahrung mit 3D-Anwendungen (38%). Eine direkte Abhängigkeit lässt sich jedoch durch die durchgeführte Sensitivitätsanalyse ausschließen.

Eine weitere Hypothese, die sich bereits aus dem Namen des Kriteriums ableiten lässt, ist die Vertrautheit im Umgang mit Robotiksystemen. Während der Studie hatten einige Proband*innen großen Respekt im Umgang mit dem Knickarmroboter und sind entsprechend langsamer und vorsichtiger bei der Teleoperation vorgegangen. Proband*innen mit Erfahrung sind deutlich pragmatischer an die Aufgabenstellung herangegangen und erzielten entsprechend bessere Versuchszeiten. Die Technologievertrautheit ist nach dieser These ein relevanter, dritter Einflussparameter auf die Teleoperationsdauer.

Sortiert man die Teleoperationszeiten der Proband*innen entsprechend t_G und teilt diese in drei unterschiedliche Gruppen:

1. 10 schnellste Proband*innen
2. 22 durchschnittlich schnelle Proband*innen
3. 10 langsamste Proband*innen

lässt sich in Abbildung 7.12 deutlich der Einfluss der drei relevanten Parameter erkennen.

7.2.2 Einfluss der Visualisierung in VR

V1 beschreibt die Teleoperation ohne VR-Brille. Die Proband*innen stehen während der Versuchsdurchführung unmittelbar vor dem Demonstrator und beobachten den Knickarmroboter, die Umgebung und die zu kommissionierenden Objekte direkt mit ihren Augen. Die Steuerung wird durch den in Kapitel 5.1 beschriebenen VR-Controller mit der ermittelten Systemlatenz von 100 ms (vgl. Kapitel 7.1) durchgeführt. Der Versuch stellt einen gewissen Sonderfall dar, da lediglich die Steuerbefehle über eine Netzwerkverbindung übertragen werden.

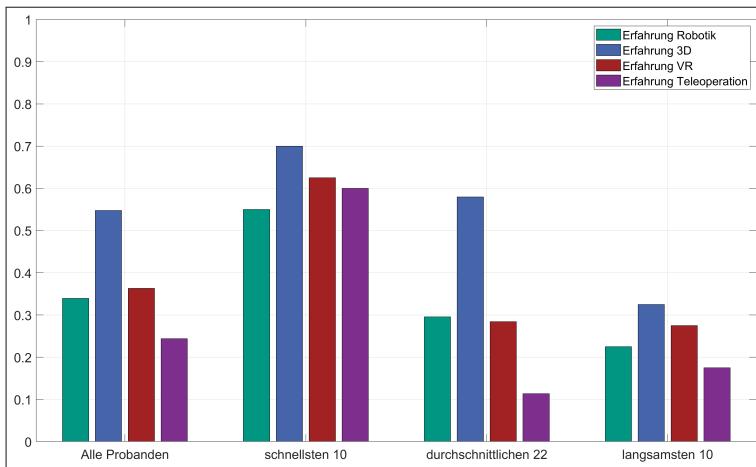


Abbildung 7.12: Selbsteinschätzung der Proband*innen in den jeweiligen Gruppen, eigene Darstellung

Die Latenz der *Sehbahn* des Menschen L_S , also die benötigte Zeit zwischen der Codierung des Bildes in der Netzhaut bis zu den ersten messbaren Impulsen in der primären *Sehrinde*, kann für gesunde Menschen mit ca. $L_S = 100 \text{ ms}$ angegeben werden [Ilg und Groß(2017)], [Plange(2017)]. Die gesamte visuelle Latenz von V1 L_{V1ges} ergibt sich somit zu:

$$L_{V1ges} = L_S = 100 \text{ ms} \quad (7.6)$$

V2 beschreibt die Teleoperation mit VR-Brille und soll den Einfluss der Visualisierung mit Hilfe einer VR-Brille bei der Teleoperation untersuchen. Die gesamte visuelle Latenz L_{V2ges} , setzt sich aus der in V1 beschriebenen Latenz der Sehbahn L_S , der Latenz der Datenaufnahme L_A , der Latenz der Datenübertragung (Transfer) L_T sowie der Darstellungslatenz auf Seite des Teleoperators L_D zusammen.

Die Latenz der Teleoperationskamera (siehe Kapitel 5.1) nimmt Bilder mit einer Frequenz von 30 Hz auf und führt max. zu einer zusätzlichen Latenz von 33 ms . Die interne Datenverarbeitungsgeschwindigkeit der Teleoperationskamera wird mit ca. 60 ms für das Farbbild und mit ca. 40 ms für das Tiefenbild

bei den in dieser Arbeit verwendeten Auflösungen angegeben [wes b(2020)]. Die Latenz der Datenaufnahme kann resultierend mit $L_A = 90ms$ angegeben werden.

Die Latenz der Datenübertragung (Werte für Kompression, Übertragung und Dekompression aus Abschnitten 5.2.3, 5.2.4 und 5.2.5) summiert sich auf $L_T = 10ms$.

Eine exakte Angabe der Darstellungslatenz auf Seite des Teleoperators L_D für das verwendete VR-Headset konnte leider nicht ermittelt werden. Allerdings existieren zwei Veröffentlichungen, die eine Latenz für die Darstellung in VR bestimmen: Raaen [Raaen und Kjellmo(2015)] untersuchte im Jahr 2015 die Darstellungslatenz an den ersten Prototypen der Oculus Rift [DJSCOE, Vile - Parle (W), Mumbai und Rajesh Desai et al.(2014)] und gibt diese mit ca. 40 ms für das dev kit2 an. Die Untersuchungen von Gruen [Gruen et al.(2020)Gruen, Ofek, Steed, Gal, Sinclair, und Gonzalez-Franco] behandeln zwar das in dieser Arbeit verwendete Headset, jedoch wird die Latenz der eingebauten Kameras bis zur Darstellung auf den Bildschirmen gemessen und mit ca. 80 ms angegeben. Unter Betracht der deutlichen technologischen Entwicklung des verwendeten Headsets im Vergleich zum Oculus Rift dev kit2 und den Ergebnissen von Gruen, wird die Darstellungslatenz auf $L_D = 20ms$ geschätzt.

Die gesamte visuelle Latenz für V2 ergibt

$$LV_{2ges} = L_A + L_T + L_D + L_S = 220ms \quad (7.7)$$

und ist somit ca. 120 ms höher als ohne VR-Headset.

Neben der Latenz kommen weitere Einschränkungen durch die Verwendung einer VR-Brille sowie der verbauten Sensorik (vgl. Kapitel 5.1) hinzu:

Während der kombinierte Öffnungswinkel der menschlichen Augen mit ca. 180° [Strasburger und Pöppel(2002)] angegeben wird, bietet die verwendete VR-Brille lediglich einen effektiven Öffnungswinkel von ca. 120°-130° [Angelov et al.(2020)Angelov, Petkov, Shipkovenski, und Kalushkov]. Der Teleoperator erhält also bereits durch die Verwendung einer VR-Brille ein deutlich engeres Sichtfeld, obwohl die verwendete VR-Brille zum Zeitpunkt der Untersuchung die bestmögliche Option bietet. Dieser Effekt wird durch den limitierten Öffnungswinkel von 75° der Teleoperationskamera weiter verstärkt.

Zwar werden die relevanten Bereiche innerhalb der Kiste vollständig durch die kolorierte Punktewolke abgedeckt, allerdings sind Bewegungen oberhalb der Kiste nicht vollständig einzusehen und der Teleoperator muss sich vollständig auf den digitalen Zwilling verlassen.

Zusätzlich erlaubt die Verwendung einer fest installierten Teleoperationskamera lediglich die Umgebungserfassung aus einer Ansicht. Wird der Knickarmroboter zwischen Kamera und zu kommissionierendem Objekt bewegt, wird dieses teilweise oder sogar vollständig verdeckt. Die Qualität der Tiefeininformationen variiert stark in Abhängigkeit von Blickwinkel und Material des betrachteten Objekts [Tölgessy et al.(2021)Tölgessy, Dekan, Chovane, und Hubinsky]. Verzerrungen und Fehlmessungen in Abhängigkeit der Objektoberflächen müssen durch den Teleoperator ausgeglichen werden.

Quantitativ lassen sich die durchschnittliche und die minimale Teleoperationszeit von V1 und V2 miteinander vergleichen:

Metrik	V1	V2	Steigerung
Durchschnitt alle	20.6	21.7	5.3% (1.1)
Minimal alle	14.0	15.6	11.4% (1.6)
Minimal schnellsten 10	9.6	10.6	10.4% (1.0)
Minimal Langsamsten 10	18.8	21.8	15.9% (3.0)

Tabelle 7.4: Vergleich der durchschnittlichen und durchschnittlich minimalen Versuchszeit der Proband*innen bei V1 (keine VR-Brille) und V2 (mit VR-Brille). Zusätzlich wurden die 10 schnellsten und 10 langsamsten Proband*innen gesondert aufgelistet. Die Steigerung wird prozentual zwischen V2 und V1 betrachtet. Alle weiteren Angaben sind in Sekunden.

Die durchschnittlichen Versuchszeiten über alle Iterationen hinweg lassen sich für eine Analyse nur bedingt anwenden, da eine deutliche Lernkurve der Proband*innen erkennbar ist (siehe Tabelle 7.3). Das Lernverhalten wird ausführlich in Abschnitt 7.2.3 untersucht.

Betrachtet man hingegen den Unterschied der durchschnittlichen minimalen Versuchszeit aller Proband*innen bei V1 und V2, ergibt sich eine Steigerung der Teleoperationsdauer von 11.4%, bzw. von 1.6 s. Für die 10 schnellsten

Proband*innen verringert sich dieser Wert auf 10.4%, bzw. 1.0 s. Für die 10 langsamsten ist eine deutlich stärkere Beeinflussung durch die Verwendung einer VR-Brille mit 15.9%, bzw. 3.0 s zu beobachten.

Vergleicht man die gefahrenen Trajektorien von V1 und V2 (Abbildung 7.10b), fällt ein ähnliches Verhalten bei der Verwendung einer VR-Brille wie bei einer Steigerung der Latenz auf. Die Steigerung lässt sich durch die erläuterte visuelle Latenz von ca. 120 ms und technische Einschränkungen der verwendeten Hardware erklären.

An dieser Stelle sei erneut hervorgehoben, dass die Teleoperationsdauer lediglich um 1.6 s, bzw. 11.4% gestiegen ist, obwohl die Proband*innen ein virtuelles Abbild der Realität betrachten und die Visualisierung und Steuerung über eine Netzwerkverbindung verzögert wird.

7.2.3 Lernfähigkeit der Proband*innen

Während für die Auswertung des Einflusses einer Visualisierung in VR die durchschnittliche minimale Versuchszeit verwendet wurde, lässt sich durch Betrachtung der durchschnittlichen Teleoperationsdauer je Iteration von V1 und V2 die Lernfähigkeit untersuchen. Tabelle 7.5 zeigt die durchschnittliche Teleoperationsdauer je Iteration für alle Proband*innen.

Iteration	1	2	3	4	5	6
V1 (kein VR)	32.0	23.7	21.0	17.7	16.7	15.8
V2 (VR)	26.0	24.6	22.4	19.8	21.2	19.2

Tabelle 7.5: Vergleich der durchschnittlichen Teleoperationsdauer je Iteration für alle Proband*innen bei V1 und V2. Alle Angaben sind in Sekunden [s].

Eine graphische Darstellung der Daten aus Tabelle 7.5 inklusive der zugehörigen Standardabweichung findet sich in Abbildung 7.13.

Während für V1 eine beachtliche Reduzierung der Teleoperationsdauer von 51% erreicht wurde ($32.0\text{s} \rightarrow 15.8\text{s}$), steigt für die Umgewöhnung nach sechs Iterationen ohne VR auf eine Visualisierung in VR die benötigte Zeit um 64%

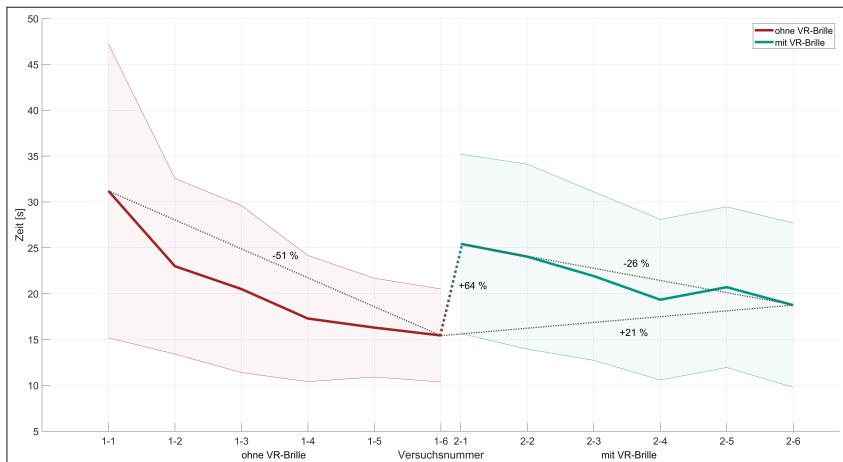


Abbildung 7.13: Lernfähigkeit aller Proband*innen, ohne VR-Brille entspricht der direkten Sicht auf den Roboter, mit VR-Brille entspricht der Sicht durch VR-Brille auf 3D-Daten. Schattierte Bereiche entsprechen der jeweiligen Standardabweichung, eigene Darstellung

(15.8s → 26.0s). Dieser Wert ist zwar immer noch etwas niedriger als die durchschnittliche Teleoperationsdauer bei der ersten Teleoperation, zeigt aber deutlich den Einfluss der in Abschnitt 7.2.2 beschriebenen Einschränkungen für eine entkoppelte Visualisierung. Die Teleoperationsdauer sinkt in den folgenden sechs Iterationen um 26% (26.0s → 19.2s) und entspricht, vergleicht man jeweils die letzte Iteration, einer Erhöhung der Teleoperationsdauer von 21% (15.8s → 19.2s).

Wie bereits in Kapitel 7.2.2 diskutiert, erreicht die Teleoperation mit Visualisierung in VR erwartungsgemäß nicht das Niveau einer Teleoperation ohne Visualisierung in VR. Bei Betrachtung der gemessenen Zeiten von V1 und V2, inklusive Standardabweichung, fällt jedoch auf, dass für V1 die Teleoperationsdauer sowie die Standardabweichung kontinuierlich sinken. Für V2 hingegen bleibt die Standardabweichung annähernd konstant und es existiert sogar eine kurzzeitige Erhöhung der durchschnittlichen Teleoperationsdauer bei der fünften Iteration. Der Einfluss der hohen Standardabweichung zeigt sich ebenfalls in der deutlich höheren Steigerung der durchschnittlichen Teleoperationsdauer

von 21%, anstelle der in Abschnitt 7.2.2 beschriebenen Erhöhung von 11.4% bei Betrachtung der durchschnittlichen, minimalen Teleoperationsdauer.

Betrachtet man die Teleoperationsdauer von V1 und V2 in der in Abschnitt 7.2.1 eingeführten Kategorisierung (Abbildung 7.14), zeigt sich ein deutlicher Unterschied im Verbesserungspotential der einzelnen Gruppen durch die unterschiedlichen Steigungen der Lernkurven. Alle drei Gruppen zeigen eine deutlich höhere Schwankung der Teleoperationsdauer für V2.

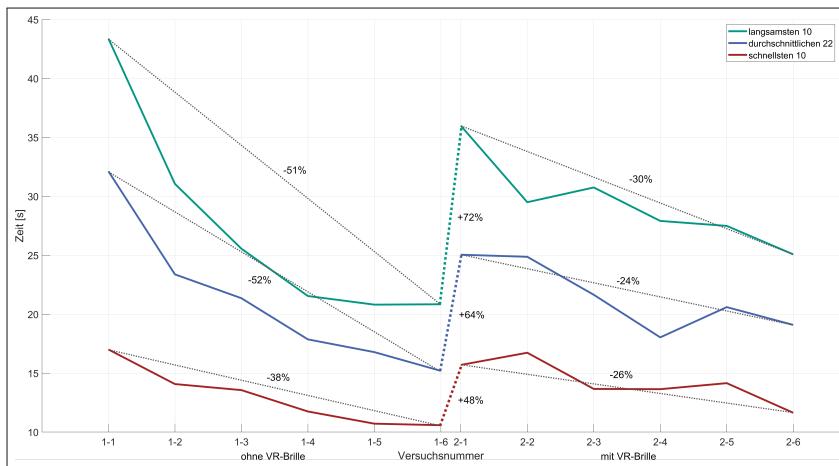


Abbildung 7.14: Lernfähigkeit der einzelnen Gruppen, ohne VR-Brille entspricht der direkten Sicht auf den Roboter, mit VR-Brille entspricht der Sicht durch VR-Brille auf 3D-Daten, eigene Darstellung

Hieraus lässt sich schließen, dass trotz Verwendung von modernster 3D-Sensorik und VR-Hardware eine Eingewöhnungsphase an die Visualisierung in VR notwendig ist. Da die erwähnten Schwankungen auch bei VR-erfahrenen Proband*innen auftreten, ist auch bei einer längeren Eingewöhnungsphase nicht mit einem besseren Ergebnis bei der Durchführung der folgenden Teleoperationen mit zusätzlicher Latenz zu rechnen.

Gruppe	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10
alle 42	0	11.55	59.78	85.89	114.05	135.68	137.74	166.26	198.16	247.14
schnellsten 10	0	10.63	54.76	80.49	97.84	128.15	154.21	171.75	260.49	277.53
durchschnittlichen 22	0	8.79	50.74	72.48	103.37	136.68	135.71	162.28	178.98	231.42
langsamsten 10	0	16.43	76.95	110.29	139.56	137.90	132.62	169.88	197.37	257.05

Tabelle 7.6: Prozentuale Verschlechterung der Teleoperationsdauer mit Bezug zu V1 (keine VR-Brille) unter Berücksichtigung der Lernfähigkeit der Proband*innen in beschriebener Gruppeneinteilung

7.2.4 Quantitative Bewertung und Modellierung des Zusammenhangs von Teleoperationsdauer und Latenz für Pick-and-Place Aufgaben

Quantitativ betrachtet lässt sich aus den ermittelten Teleoperationszeiten der Proband*innen aus Tabelle 7.3 bzw. aus der visuellen Darstellung der durchschnittlichen Teleoperationszeit (Abbildung 7.9) eine kontinuierliche Steigerung der Teleoperationszeit erkennen. Diese variiert zwischen 2,8% und 16,8% zum jeweilig vorhergehenden Versuch. Der Zusammenhang zwischen Latenz und Teleoperationszeit ist also keinesfalls linear.

Wird die Lernfähigkeit der Proband*innen innerhalb der ersten beiden Versuche berücksichtigt und jeweils die durchschnittlich minimale Zeit für V1 und V2 sowie für alle folgenden Versuche (V3 - V10) die durchschnittliche Versuchszeit herangezogen, ergeben sich die in Tabelle 7.6 angegebenen prozentualen Verschlechterungen der Teleoperationsdauer in Bezug zu V1. Abbildung 7.15 visualisiert die Werte aus Tabelle 7.6.

Das bereits vorgestellte Modell von Pongrac [Pongrac(2008)] (Abschnitt 2.4 bzw. Abbildung 2.10) stellt ursprünglich den Zusammenhang von Teleoperationsdauer und Latenz mit einer dreistufigen Funktion dar. Jedoch konnte Pongrac in ihrer Studie lediglich eine zweistufige Funktionen für teleoperierte Greifaufgaben ableiten. Diese besteht aus einer linearen Funktion mit einer Steigung von 0 für Latenzen bis 75 ms und einer quadratischen Funktion für Latenzen $> 75\text{ ms}$. Die zweiteilige Funktion erreicht ein Bestimmtheitsmaß von $R^2 = 89.9\%$. Im Vergleich zu der hier vorgestellten Studie ist die zu erledigende Aufgabe leichter, da nur ein Greifvorgang ohne Ablegen untersucht wurde.

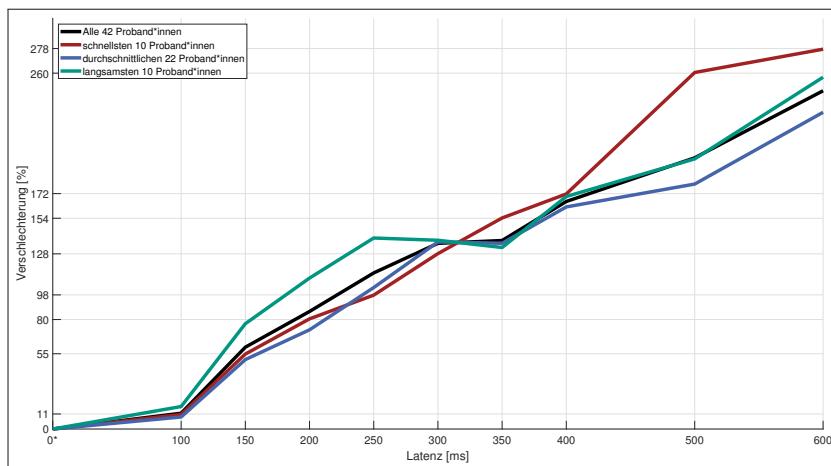


Abbildung 7.15: Einfluss der Latenz auf drei unterschiedliche Gruppen der Proband*innen, eigene Darstellung

Datenbetrachtung Bei Betrachtung der im Rahmen dieser Dissertation ermittelten Daten (siehe Abbildung 7.15) ist die hohe Verschlechterung der Leistungsfähigkeit von 100ms zu 150ms sehr auffällig. Alle Proband*innen haben an dieser Stelle bereits 12 Teleoperationen ohne künstlich wirkende Latenz durchgeführt und sind mit System und Steuerung vertraut. Ebenfalls auffällig ist eine Abflachung der Steigung im Bereich 250ms bis 350ms gefolgt von einer erneuten Steigung für Latenzen $> 350\text{ms}$.

Die Abflachung kann durch die These eines Steuerungswechsels des Teleoperators erklärt werden. Während bei niedrigen Latenzen eine *intuitiv-direkte Steuerung* noch möglich ist, findet für Latenzen ab 250ms ein Wechsel zu einer *Bewegen-Warten Steuerung* statt. Der Steuerungswechsel bei 300ms wird ebenfalls durch die höchste Varianz der Leistungsfähigkeit aller Proband*innen untermauert (siehe Abbildung 7.9). Die nächste Latenzstufe bei 350ms kann als Gewöhnung an die neue Steuerstrategie gewertet werden und gleicht entsprechend die wirkende Latenzsteigerung aus.

Regressionsanalyse Eine Regression für Latenzen $< 100ms$ würde auf Grund von mangelnder Datenlage kein belastbares Ergebnis liefern, da bereits die Systemlatenz zwischen Steuersignal und Endeffektor des Roboters $100ms$ beträgt. Die Berechnung der Regression bezieht sich somit nur auf Latenzen $> 100ms$. Die bereits diskutierten Erkenntnisse über den Einfluss einer Visualisierung in VR (vgl. Abschnitt 7.2.2) lassen jedoch vermuten, dass bereits leichte Veränderungen des realen Abbilds (Latenzen, 3D-Daten, etc.) einen Einfluss auf die Teleoperationsdauer haben.

Um herauszufinden inwieweit eine Unterteilung der Regressionsfunktion und der Polynomgrad der Funktion die Güte der Regression beeinflussen, wurde ein Optimierungsverfahren entwickelt, das die Daten in bis zu drei Segmente unterteilen kann. Je Segment wird eine Regressionsfunktion mit den Polynomgraden $P \in [1, 2]$ berechnet. Die Regression kann für jedes Segment somit je nach Wert der Kostenfunktion 1. Ordnung oder 2. Ordnung sein.

Als Kostenfunktion wird die mittlere quadratische Abweichung (engl. mean squared error: MSE) verwendet (siehe Formel 7.8).

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - f(x_i))^2 \quad (7.8)$$

Handelt es sich um zwei oder drei Segmente, wird die Berechnung der mittleren quadratischen Abweichung um eine Gewichtsfunktion ergänzt, die die Anzahl der Messpunkte innerhalb des Segments berücksichtigt. Die Kostenfunktion ist somit eine gewichtete mittlere quadratische Abweichung (engl: weighted mean squared error: WMSE) (siehe Formel 7.9).

$$WMSE = \frac{1}{n} \sum_{i=1}^n w_i (y_i - f(x_i))^2 \quad (7.9)$$

Damit lineare Abschnitte nicht auf Grund eines geringfügig besseren MSE-Wertes durch Polynome 2. Ordnung ersetzt werden, wird die lineare Regression präferiert gewählt, solange der MSE- bzw. WMSE-Wert des jeweiligen Abschnitts sich um weniger als 5% erhöht.

Die ermittelten Regressionsparameter und Segmente sind in Abbildung 7.16 graphisch dargestellt. Die ermittelten Funktionsparameter sind im Folgenden für jedes Segment einzeln gelistet:

Regressionsparameter für 1 Segment, $MSE = 83.85$:

$$T(L) = \begin{cases} -0.0003L^2 + 0.63L - 35.21 & 100ms < L \end{cases} \quad (7.10)$$

Regressionsparameter für 2 Segmente, $WMSE = 11.43$:

$$T(L) = \begin{cases} -0.001L^2 + 1.3L - 100.21 & 100ms < L \leq 350ms \\ +0.421L - 7.40 & 350ms < L \end{cases} \quad (7.11)$$

Regressionsparameter für 3 Segmente, $WMSE = 3.6$:

$$T(L) = \begin{cases} -0.001L^2 + 1.2L - 90.8 & 100ms < L \leq 300ms \\ +0.005L^2 - 3.4L + 678.9 & 350ms < L \leq 400ms \\ +0.001L^2 - 0.4L + 209.5 & 400ms < L \end{cases} \quad (7.12)$$

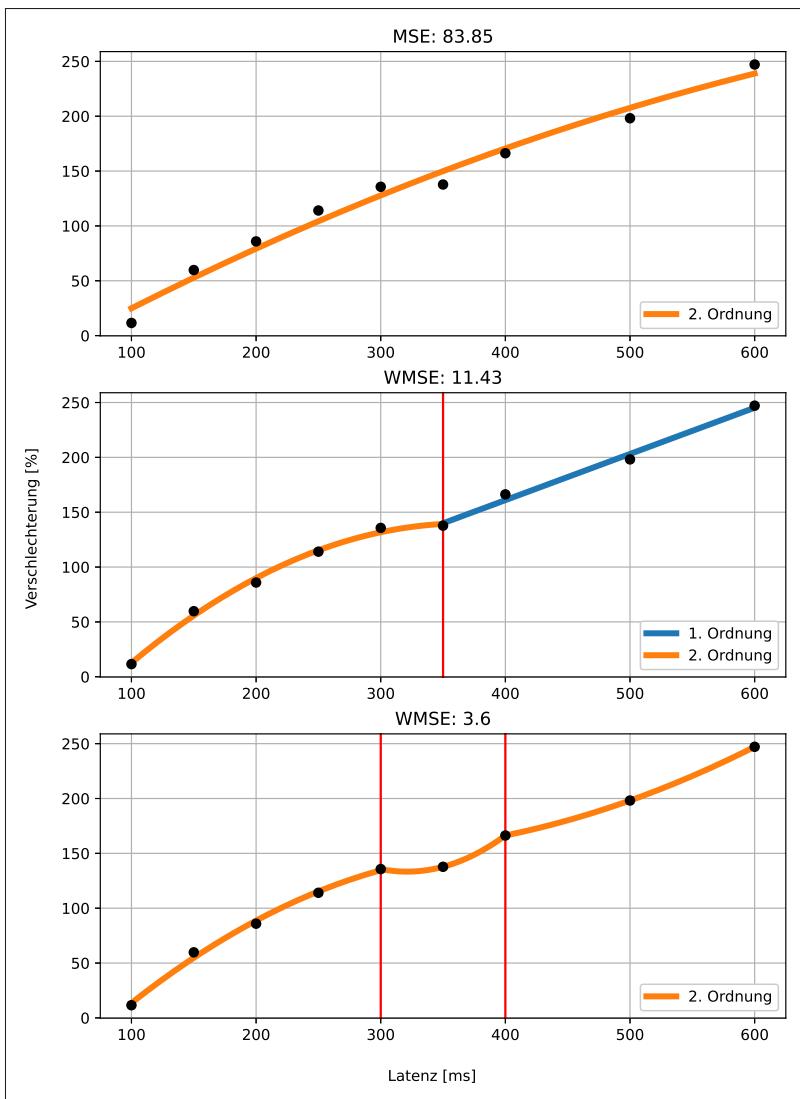


Abbildung 7.16: Ergebnis der polynomialen Regressionsoptimierung der prozentualen Verschlechterung der Teleoperationsdauer in Abhängigkeit der wirkenden Latenz.
 (a) 1 Segment (b) zwei Segmente (c) drei Segmente, eigene Darstellung

Auswertung der Regressionsoptimierung Die erhaltenen Regressionsergebnisse sollen auf Basis von insgesamt vier Kriterien beurteilt werden:

1. Je höher die Latenz, desto höher die prozentuale Verschlechterung (siehe Kapitel 7.2.1)
2. Wechsel von *intuitiv-direkter Steuerung* zu *Bewegen-Warten Steuerung* deutlich erkannbar (siehe Abschnitt 7.2.4 und Kapitel 2)
3. Möglichst niedriger MSE-Wert
4. keine Überangleichung

Während sich Kriterium 1 und Kriterium 2 auf bereits diskutierte Erkenntnisse des erhaltenen Datensatzes stützen, beschreibt Punkt 3 die Güte der gefundenen Regression. Punkt 4 motiviert dazu, eine Regression mit einer möglichst kleinen Anzahl an Segmenten zu wählen. Hierdurch soll eine Überangleichung (engl. overfitting) an die aufgenommenen Daten vermieden werden. Die Anwendung der vier Kriterien auf die erhaltenen Regressionsergebnisse ist in Tabelle 7.7 dargestellt:

Segmente	Kriterium	1	2	3	4
1		✗	✗	✗	✓
2		✓	✓	✓	✓
3		✓	✓	✓	✗

Tabelle 7.7: Anwendung der vier Kriterien (siehe 7.2.4) auf die Regressionsergebnisse

Bei Betrachtung der Ergebnisse der Regressionsoptimierung in Abbildung 7.16 fällt zunächst auf, dass eine Unterteilung in drei Segmente zu einer Überanpassung (engl. overfitting) der Daten führt. Die Kurven der einzelnen Segmente passen sich sehr genau an den Verlauf der Punkte an, was zu einem niedrigen WMSE-Wert führt. Jedes Segment wird durch ein Polynom 2. Ordnung beschrieben, die Segmente bestehen jedoch zum Teil aus lediglich drei Punkten. Diese Variante ist ebenfalls auszuschließen.

Im Gegensatz zu einer Unterteilung in drei Segmente gehen bei der Verwendung von nur einem Segment wichtige Details verloren. Hierzu gehören neben den vier Kriterien auch noch die stark ansteigende Verschlechterung von $100ms$ auf $150ms$. Eine negative Steigung des Polynoms bedeutet einen Rückgang der Verschlechterung bei steigenden Latenzen. Diese Variante ist somit definitiv auszuschließen.

Die polynomiale Regression mit einer Unterteilung in zwei Segmente liefert entsprechend der vier beschriebenen Kriterien das beste Ergebnis. Alle vier Kriterien sind hierbei erfüllt. Der Kurvenverlauf wird im ersten Segment durch ein Polynom 2. Ordnung und im zweiten Segment durch ein Polynom 1. Ordnung (lineare Funktion) beschrieben. Ein Polynom 2. Ordnung im zweiten Segment würde den WMSE-Wert um lediglich 2.4% verbessern. Ein linearer Verlauf stellt jedoch unter Berücksichtigung der Bewegen-Warten Strategie sowie des ebenfalls sehr guten WMSE-Wertes das plausibelste Ergebnis dar.

Für einen realen Einsatz von Teleoperationssystemen liegt somit eine kritische Ende-zu-Ende Latenz ab der eine Verschlechterung entsprechend der quadratischen Funktion $T(L) = -0.001L^2 + 1.3L - 100.21$ zu erwarten ist bereits unter $150ms$. Ab einer Latenz von $350ms$ ist somit nur mit einer linearen Verschlechterung entsprechend der linearen Funktion $T(L) = +0.421L - 7.40$ zu rechnen (entsprechend Gleichung 7.11).

Abgrenzung zu den Ergebnissen von Pongrac [Pongrac(2008)] Zwischen den Ergebnissen der vorliegenden Dissertation sowie der Studie von Pongrac [Pongrac(2008)] besteht eine deutliche Diskrepanz (siehe Abbildung 7.17).

Die von Pongrac [Pongrac(2008)] ermittelte Regressionskurve besteht aus einem einheitlichen, polynominalen Verlauf für Latenzen über $75ms$. Die in dieser Arbeit ermittelten Teleoperationszeiten und diskutierten Effekte lassen sich hiermit nicht abbilden. Zum einen zeigt die Regressionskurve von Pongrac einen flacheren Verlauf für Latenzen zwischen $100ms$ und $200ms$ als die im Rahmen dieser Arbeit ermittelte Regressionskurve. Zum anderen wird der Effekt eines Steuerungswechsels des Teleoperators bei $350ms$ nicht abgebildet (siehe Abschnitt 7.2.4 und Kapitel 2). Der quadratische Kurvenverlauf suggeriert zudem einen Rückgang der Leistungsverschlechterung bei weiter steigenden Latenzen.

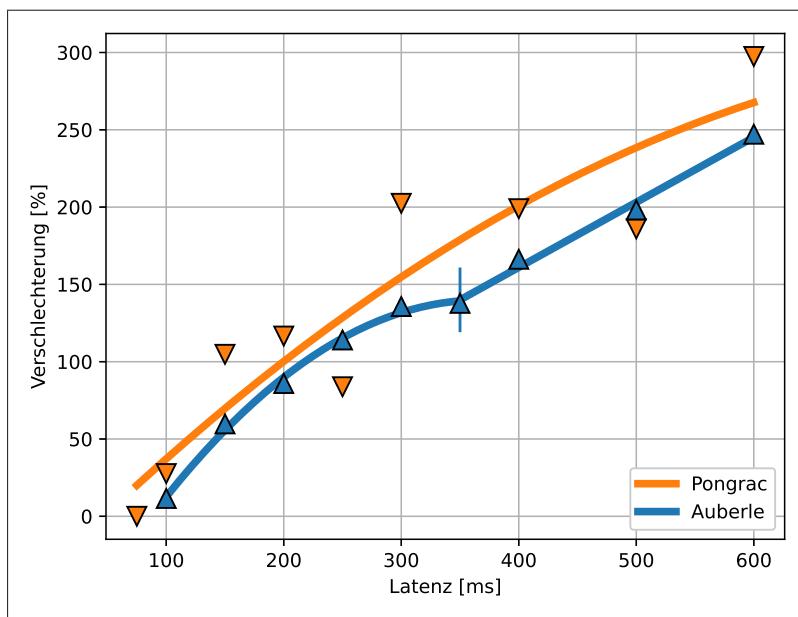


Abbildung 7.17: Vergleich der Regressionsergebnisse von Pongrac [Pongrac(2008)] (adaptiert auf prozentuelle Verschlechterung) und der im Rahmen dieser Arbeit ermittelten Regressionskurve, eigene Darstellung

Die Schwierigkeit der zu absolvierenden Aufgabe, erhebliche technische Fortschritte bei VR-Komponenten und Sensorik und die damit verbundene Immersionssteigerung sind leider nur schwer messbare Kriterien, die mit Sicherheit einen Einfluss auf die Unterschiede in den gemessenen Ergebnissen haben. Da Pongrac die tatsächlich wirkenden Systemlatenzen jedoch nicht vollumfänglich beschrieben hat, ist es möglich, dass die Latenzen höher waren als angegeben. In der Tat ist ein Rückgang der Verschlechterung für Latenzen zwischen 300ms und 500ms in den von Pongrac ermittelten Daten zu erkennen, also ca. 150ms später als in dem in dieser Arbeit vorgestellten Kurvenverlauf.

Weiterhin wurde für die in dieser Arbeit vorgestellten Studie keine Unterteilung der Proband*innen durchgeführt. In der Studie von Pongrac wurde eine Aufspaltung in drei Gruppen gemacht, wobei jede Gruppe mit unterschiedlichen Latenzen testete. In Anbetracht der ermittelten hohen Varianz zwischen

den schnellsten und langsamsten Proband*innen führt diese Aufteilung mit hoher Wahrscheinlichkeit zu einer Verschiebung einzelner Messpunkte.

Nach bestem Wissen ist die durchgeführte Studie für die Anwendung von Kommissionieraufgaben an Pick-and-Place Robotern einzigartig und stellt zugleich die größte Studie mit 42 Proband*innen und 1258 durchgeführten Teleoperationen zur Untersuchung des Einflusses von Latenz dar. Die Kombination aus leichter und schwieriger Aufgabe ohne Unterbrechung simuliert eine reale Aufgabenstellung und führt zu einer mehrmaligen Kombination aus *freier Bewegung* und *Kontaktaufgabe*.

Resultierend stellt die ermittelte zweistufige Funktion sowie der diskutierte Einfluss einer Visualisierung in VR einen neuen Stand der Technik dar und beantwortet **Fragestellung 5: Wie ist der Einfluss von Latenz zwischen Teleoperator und CPS?**.

7.3 Lernen durch Teleoperation

Der folgende Abschnitt verbindet alle vorgestellten Module nach dem Konzept für *kontinuierliches Lernen aus Teleoperation*, das in Kapitel 4 vorgestellt wurde. Der Versuchsaufbau und die Versuchsdurchführung sind als Funktionsdemonstration des erarbeiteten Frameworks für die Anwendung mit einem Kommissionierroboter zu verstehen.

Grundsätzlich sind mehrere Versuchsaufbauten zur Demonstration der Lernfähigkeit denkbar. Wie bereits in Kapitel 4.2 und 4.6 beschrieben, ist hierfür nahezu jede automatisierte Funktion geeignet, solange diese vollständig oder in Teilen dem Bereich des *Maschinellen Lernens* zuzuordnen ist. Nachfolgend soll jedoch ausschließlich auf die Lernfähigkeit der verwendeten Objekterkennung (vgl. Kapitel 6.1) eingegangen werden.

Die Wahl der Objekterkennung für eine Funktionsdemonstration hat drei Hauptgründe: Zunächst eine einfache Interpretierbarkeit der Ergebnisse sowie die große Relevanz des schnellen Einlernens von neuen Artikeln für Kommissionierroboter. Darauf hinaus ist die Generierung von Lerndaten für die verwendete Objekterkennung (Abschnitt 6.2) nicht trivial, da diese nicht instanzbasiert, sondern szenenbasiert arbeitet. Es müssten also alle Objekte in einer Szene annotiert werden, um verwendbare Lerndaten zu generieren.

7.3.1 Versuchsaufbau

Die Lernfähigkeit der Objekterkennung sowie der Einfluss der Hilfestellung durch Teleoperation soll anhand der automatisierten Kommissionierung von fünf handelsüblichen, rechteckigen und unterschiedlich texturierten Objekten stattfinden. Die Objekte sind in Abbildung 7.18 dargestellt. Der entwickelte Demonstrator (vgl. Kapitel 5.1) verfügt über zwei Kleinladungsträger (KLT), von denen stets eine als Quelle und die andere als Ziel genutzt wird. Ein Kommissionierauftrag besteht somit aus fünf Pick-and-Place Aufgaben.



Abbildung 7.18: Eingelernte Referenzbilder der zu greifenden Objekte, eigene Darstellung

Die *Objekterkennung* sowie die *Greifpunktterkennung* haben zu Beginn des Versuchs die fünf Objekte noch nie zuvor gesehen. Beide Module werden jedoch mit vortrainierten Gewichten eingesetzt: Die Objekterkennung mit OS2D ist auf Basis des GroZi-3.2K Datensatzes [George und Floerkemeier(2014)] mit den Labeldaten von Osokin [Osokin et al.(2020)Osokin, Sumin, und Lomakin] auf 8921 Objekteninstanzen in 1063 Klassen trainiert worden. Die Greifpunktterkennung wurde entsprechend des im Paper vorgestellten Datensatzes [Hopfgarten et al.(2020)Hopfgarten, Auberle, und Hein] vortrainiert und wird mit DeepLab V3+ [Chen et al.(2018)Chen, Zhu, Papandreou, Schroff, und Adam] als Backbone eingesetzt.

Zunächst soll das Robotiksystem 30 Kommissionieraufträge (150 Pick-and-Place Aufgaben) automatisiert durchführen. Anschließend findet ein Training der Objekterkennung auf Basis des generierten Datensatzes statt. Nachfolgend sollen 150 weitere Kommissionieraufträge (750 Pick-and-Place Aufgaben) zur Evaluation durchgeführt werden.

Eine konkrete Verknüpfung der Module *Automatisierte Funktion*, *Anomalieerkennung*, *Lerndatengenerierung* und *Lernen* des erarbeiteten Frameworks (4) für einen Kommissionierroboter ist in Abbildung 7.19 dargestellt.

Ausgangspunkt ist stets ein neuer Kommissionierauftrag, der in Form einer json-Datei lediglich die Artikelnamen beinhaltet. Davon ausgehend überprüft die *Objekterkennung*, als Teil der *Automatisierten Funktion*, ob der zu kommissionierende Artikel bereits bekannt ist. Falls nein, wird unmittelbar ein Teleoperator um Hilfe gebeten, das neue Objekt entweder durch Einzeichnen einer einzelnen Boundingbox (Teleoperationsstufe 1/2) einzulernen oder mit Hilfe der entwickelten, direkten Steuerung (Teleoperationsstufe 3/4) eine Kommissionierung des Artikels durchzuführen.

Anschließend versucht die *Objekterkennung* den Artikel automatisiert zu lokalisieren. Im Fehlerfall wird erneut ein Teleoperator um Hilfe gebeten. Die *Greifpunkterkennung* geht analog zur *Objekterkennung* vor. Der Prozess wird fortlaufend von der *Anomalieerkennung* überwacht, ob berechnete Aufnahme- bzw. Ablagepositionen plausibel erscheinen und ob ein geplanter Griff erfolgreich durchgeführt wurde.

Ist eine Hilfestellung durch die Teleoperationsstufe 1/2 nicht zielführend, ist eine höhere Teleoperationsstufe (direkte Steuerung mit Stufe 3/4) notwendig. Des Weiteren prüft die *Anomalieerkennung* am Ende jedes Kommissionierauftrags, ob alle Artikel aus der Quelle entnommen wurden. Sollte dies nicht der Fall sein, wird direkt eine Teleoperation der Stufe 3/4 angefordert.

Die *Lerndatengenerierung* speichert annotierte Bilddaten direkt aus dem Einlernvorgang oder aus der Hilfestellung eines Teleoperators mit Teleoperationsstufe 1/2. Darüber hinaus werden durch DORI (siehe Kapitel 6.5) für die Teleoperationsstufen 3/4 sowie für automatisierte Kommissionierungen annotierte Bilddaten generiert. Der dabei entstehende Datensatz wird für ein späteres Training der *Automatisierten Funktion* gespeichert.

Eine Umrechnung der Bildkoordinaten zwischen Automationskamera und Teleoperationskamera des Demonstrators findet auf Basis der durchgeführten extrinsischen Kalibrierung und den ermittelten, intrinsischen Parametern der beiden Kamerä statt.

Alle prozessrelevanten Daten werden bereits während der Kommissionierung innerhalb einer Datenbank abgespeichert.

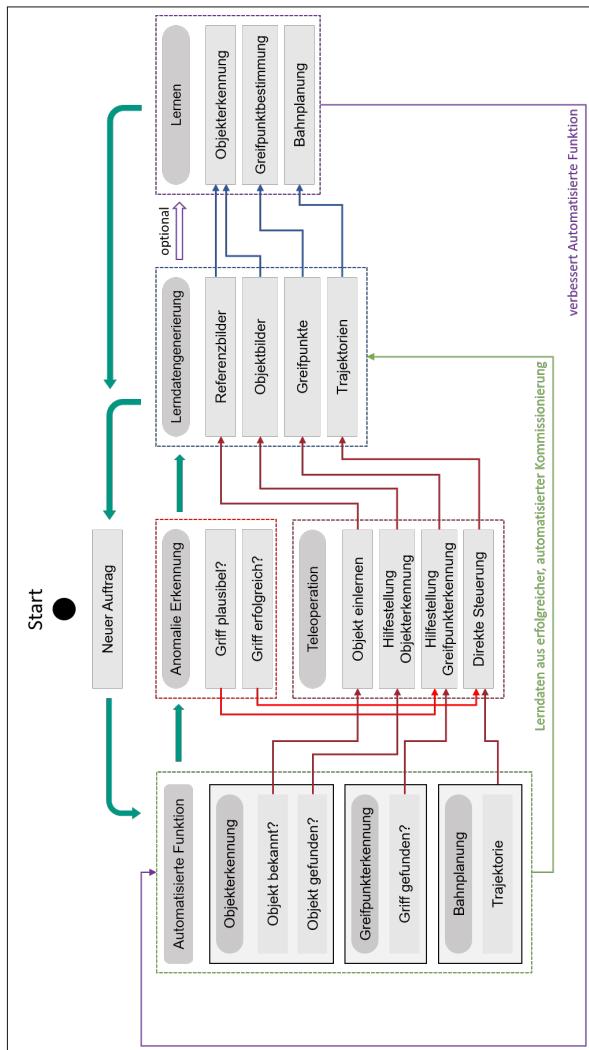


Abbildung 7.19: Anwendung des in Kapitel 4 vorgestellten Konzepts für kontinuierliches Lernen durch Teleoperation mit einem Kommissionierroboter, eigene Darstellung

7.3.2 Versuchsdurchführung und Versuchsauswertung

Entsprechend des vorgestellten Versuchsaufbaus (Abschnitt 7.3.1) wurden zunächst 30 Kommissionieraufträge mit 150 Pick-and-Place Aufgaben durchgeführt. Eine Auswertung erfolgt auf Basis der automatisierten Erfolgsquote, also der Anzahl an bereits vollständig automatisierten Artikeln im Vergleich zur Gesamtzahl, der Fehlerquote der Objekterkennung, der Fehlerquote der Greifpunktbestimmung sowie der Teleoperationsquote für die Stufen 1/2 und die Stufen 3/4. Der Einlernvorgang (inklusive Kommissionierung) wird als Teleoperation der Stufe 1/2 gezählt, die eingelernten Bilder sind in Abbildung 7.21 dargestellt. Die Ergebnisse sind in Tabelle 7.8 sowie graphisch in Abbildung 7.20 dargestellt.

Metrik	absolut	prozentual
Erfolgreiche Pick-and-Place Aufgaben	150	100.0%
Automatisiert durchgeführt	109	72.7%
Fehlerquote Objekterkennung	29	19.3%
Fehlerquote Greifpunktbestimmung	4	2.7%
Teleoperationsquote Stufe 1/2	38	25.3%
Teleoperationsquote Stufe 3/4	3	2.0%

Tabelle 7.8: Auswertung der ersten 150 Pick-and-Place Aufgaben vor einem Training der Objekterkennung

Hervorzuheben ist, dass alle Pick-and-Place Aufgaben durch die Anwendung des erarbeiteten Frameworks erfolgreich durchgeführt werden konnten. Hierzu gehören die ersten 5 teleoperierten Kommissionierungen, die das Einlernen eines neuen Objektes beschreiben. Hinzu kommen weitere 36 teleoperierte Kommissionierungen, von denen 33 mit der Teleoperationsstufe 1/2 und 3 mit der Teleoperationsstufe 3/4 durchgeführt wurden. Hauptursache für das notwendige Einschreiten eines Teleoperators ist die Objekterkennung mit einer Fehlerquote von 19.3%, zu der die Fehlerquote der Greifpunktbestimmung mit 2.7% hinzu kommt.

Insgesamt konnten lediglich 72.7% (109) aller Pick-and-Place Aufgaben automatisiert durchgeführt werden. In Kombination mit dem vorgestelltem Framework konnten jedoch 100% der Kommissionierungen ohne ein Eingreifen vor Ort durchgeführt werden.

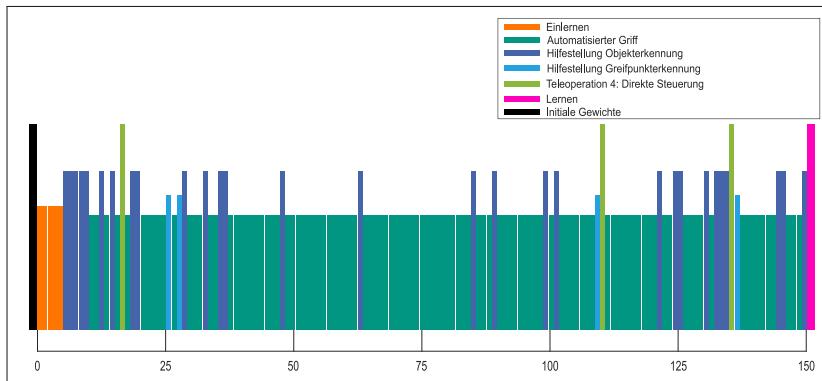


Abbildung 7.20: Ergebnis der ersten 150 Pick-and-Place Aufgaben mit fünf unbekannten Objekten, eigene Darstellung

Die Ursache für die hohe Fehlerquote der Objekterkennung von 19.3% (29) lässt sich mit Hilfe von Abbildung 7.21 erklären, die alle Hilfestellungen bei der Objekterkennung visualisiert. Die zu erkennenden Objekte ähneln sich in Form und Textur: Zwei Würfel, einer mit blauer und einer mit gelber Grundfarbe, ein Rechteck in Gelb sowie zwei in Blau-Weiß. Die Merkmalsextrahierung der Objekterkennung funktioniert reibungslos, es kommt zu keiner Fehlererkennung. Das gesuchte Objekt wird mit einem manuell gewählten Schwellwert der Erkennungsbewertung (Abschnitt 6.1) von 0.9 entweder korrekt zugeordnet oder nicht gefunden.

In Abbildung 7.21 fällt insbesondere die mangelnde Fähigkeit der Objekterkennung auf, eine Rotation des zu erkennenden Objekts mit dem vorliegenden Referenzbild abzugleichen. Dieser Effekt verstärkt sich für das kleinste der fünf Objekte mit insgesamt 11 aufgenommenen Referenzbildern mit unterschiedlichen Orientierungen. Alle aufgenommenen Referenzbilder werden mit Hilfe der entwickelten Erweiterung (vgl. Abschnitt 6.1) dem zu erkennenden Objekt zugewiesen und während der automatisierten Objekterkennung berücksichtigt.



Abbildung 7.21: Durch Teleoperation hinzugefügte Referenzbilder während der ersten 150 Kommissionierungen auf Grund von Erkennungsfehlern der Objekterkennung, eigene Darstellung

Auf Basis der eingezeichneten Objektpositionen mit der Teleoperationsstufe 1/2 sowie der direkten Steuerung mit Teleoperationsstufe 3/4 und der automatisierten Lerndatengenerierung DORI (vgl. 6.5) wurde ein visueller Lerndatensatz mit insgesamt 300 annotierten Bildern erstellt. Ein Beispiel der annotierten Daten der beiden Kameras ist in Abbildung 7.22 dargestellt.

Die Objekterkennung wurde mit dem während des Betriebs erstellten Datensatzes trainiert¹⁾ und ein weiterer Versuch mit 150 Kommissionieraufträgen bzw. 750 Pick-and-Place Aufgaben zu Evaluationszwecken durchgeführt. Bis auf die zu Beginn der ersten Versuchsreihe eingelernten Referenzbilder der Objekterkennung wurden alle durch Teleoperation hinzugefügten Referenzbilder entfernt.

Das Ergebnis ist in Tabelle 7.9 sowie graphisch in Abbildung 7.23 dargestellt.

Erneut konnten alle 30 Kommissionieraufträge und damit 750 Pick-and-Place Aufgaben zu 100% durch Anwendung des Frameworks aus Kapitel 4.6 erfolgreich durchgeführt werden.

¹ Die Trainingsparameter befinden sich im Anhang 9.1

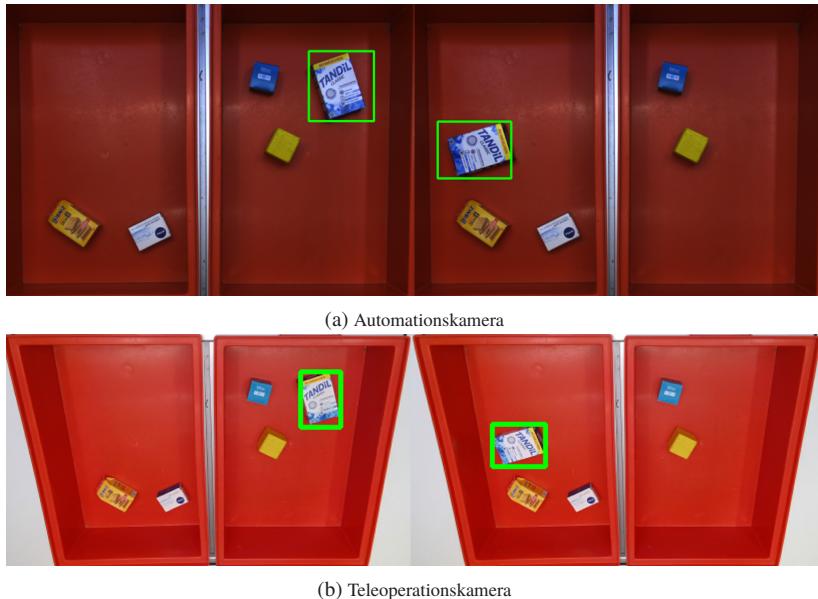


Abbildung 7.22: Visualisierung der generierten Lerndaten nach einer erfolgreich ausgeführten Kommissionierung. Diese wurde entweder vollständig automatisiert oder mit Hilfe eines Teleoperators durchgeführt. Für das Training wurden 150 Bildpaare pro Kamera aufgenommen und automatisiert annotiert.

Metrik	absolut	prozentual	vorher [%]	Veränderung
Erfolgreiche Pick-and-Place Aufgaben	750	100.0%	100.0%	±0.0%
Automatisiert durchgeführt	731	97.5%	72.7%	+24.8%
Fehlerquote Objekterkennung	2	0.3%	19.3%	-19.1%
Fehlerquote Greifpunktbestimmung	15	2.0%	2.7%	-0.7%
Teleoperationsquote Stufe 1/2	17	2.3%	25.3%	-23.1%
Teleoperationsquote Stufe 3/4	2	0.3%	2.0%	-1.7%

Tabelle 7.9: Auswertung der 750 Pick-and-Place Aufgaben nach einem Training der Objekterkennung und Vergleich zur vorherigen Leistungsfähigkeit

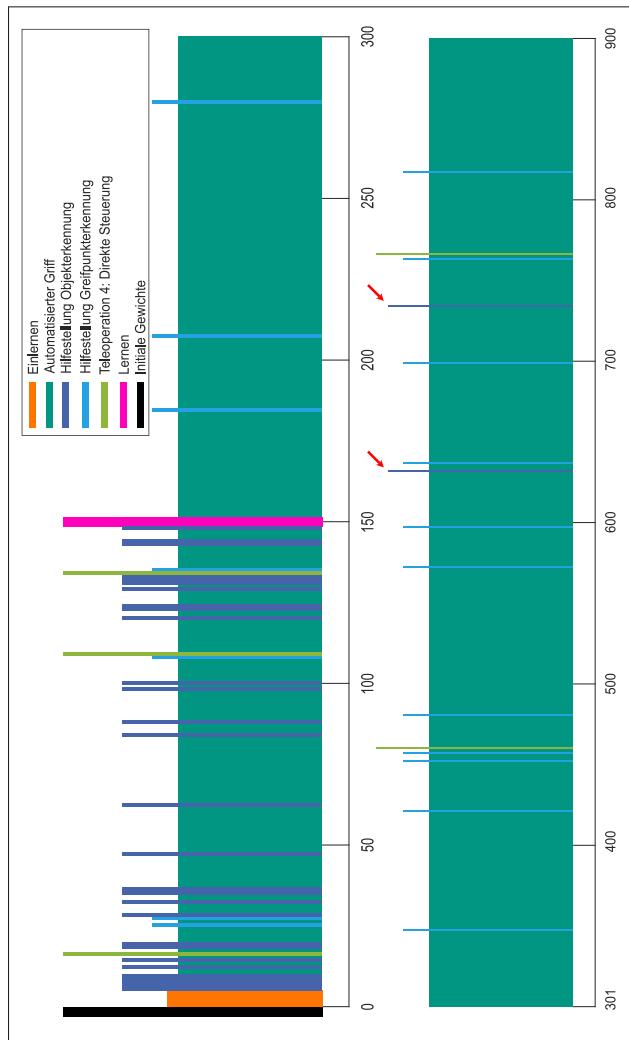


Abbildung 7.23: Leistungsanalyse von allen durchgeföhrten Pick-and-Place Aufgaben vor und nach dem Training. Nach dem Training sind lediglich zwei Eingriffe des Teleoperators auf Grund der Objekterkennung notwendig (rote Pfeile), eigene Darstellung

Im Gegensatz zur Leistungsfähigkeit vor einem Training der Objekterkennung konnte die Anzahl an automatisiert durchgeführten Kommissionierungen von 72.7% auf 97.5% signifikant erhöht werden. Erwartungsgemäß ist die Fehlerquote der Greipunktbestimmung mit einer Veränderung von lediglich 0.7% annähernd konstant geblieben, da dieses Modul nicht Teil des lernenden Versuchsaufbaus ist. Ausschlaggebend für die deutliche Reduzierung der benötigten Teleoperationen von 25.3% auf 2.3% für Teleoperationen der Stufe 1/2 sowie von 2.0% auf 0.3% für die Stufen 3/4 ist die trainierte Objekterkennung. Mit lediglich 2 Fehlerkennungen von 750 Pick-and-Place Aufgaben wird die Fehlerquote von 19.3% auf 0.3% reduziert.

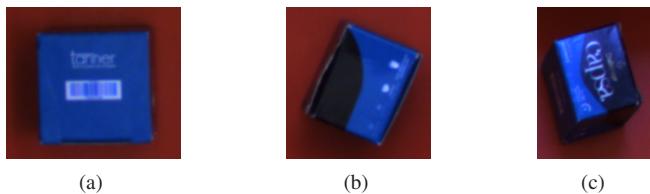


Abbildung 7.24: Benötigte Hilfestellung nach Trainings der Objekterkennung. Das zu erkennende Objekt hat sich während des Greifprozesses gewendet. (a): Referenzbild, (b): Hilfestellung 1, (c): Hilfestellung 2, eigene Darstellung

Bei Betrachtung der 2 benötigten Hilfestellungen der Objekterkennung (Abbildung 7.24) fällt auf, dass es sich nicht um eine Fehlerkennung, sondern um eine bisher nicht sichtbare (und nicht eingelernte) Oberfläche des zu erkennenden Objekts handelte. Das zu erkennende Objekt hat sich während des Greifprozesses gewendet. Die neuen Ansichten wurden entsprechend der vorgestellten Erweiterung (Abschnitt 6.1) dem Objekt zugeordnet und eine automatisierte Kommissionierung konnte ohne weitere Zwischenfälle fortgesetzt werden.

Die erhaltenen Ergebnisse zeigen, dass eine Hilfestellung durch Teleoperation zu einer unmittelbaren Leistungssteigerung des CPS führen. Alle Pick-and-Place Aufgaben konnten bereits vor einem Training der Objekterkennung unter Anwendung des entwickelten Frameworks erfolgreich durchgeführt werden. Darüber hinaus wurde eine mittelbare Leistungssteigerung des CPS mit 97.5% Erfolgsquote für eine automatisierte Kommissionierung durch Anwendung der generierten Lerndatengenerierung erreicht.

Abschließend kann ***Fragestellung 6: Kann durch das entwickelte Framework die Leistungsfähigkeit eines CPS mittelbar und unmittelbar verbessert werden?*** mit **Ja** beantwortet werden.

8 Zusammenfassung und Fazit

Anhand der Beantwortung von sechs forschungsleitenden Fragestellungen konnte gezeigt werden, wie ein Zusammenspiel von Mensch und Maschine durch Teleoperation die Leistungsfähigkeit von modernen cyber-physischen Systemen (CPS) steigert. Das entwickelte Konzept für kontinuierliches Lernen von CPS durch Teleoperation wurde zunächst allgemein formuliert. Anschließend wurde die Hilfestellung des Menschen durch Teleoperation in vier Teleoperationsstufen kategorisiert. Mit einer Proband*innenstudie wurden relevante Randbedingungen der Teleoperation identifiziert und die Funktionalität des entwickelten Konzepts anhand eines Kommissionierroboters demonstriert.

Die Ergebnisse der Proband*innenstudie zeigen, dass eine Teleoperation von CPS auch für unerfahrene Bediener*innen intuitiv umzusetzen ist. Erfahrung mit 3D-Anwendungen, VR-Anwendungen und Automatisierungstechnik bzw. Robotik begünstigen die Teleoperationsdauer. Der Einfluss von Latenz auf die Teleoperationsdauer wurde ausführlich untersucht und mit einer dreistufigen Funktion beschrieben. Der beobachtete Leistungsabfall durch eine Visualisierung in VR zeigt zum einen die Wichtigkeit einer realitätsnahen Abbildung und zum anderen das Verbesserungspotential von VR-Hardware und 3D-Sensorik.

Die vorgestellten Module zur Untersuchung der Lernfähigkeit stellen das Beispiel einer Umsetzung des entwickelten Konzepts dar. Es konnte gezeigt werden, dass durch Anwendung des Konzepts ein Kommissionierroboter unmittelbar mit einer bisher unbekannten Aufgabenstellung zurechtkommt. Neue Objekte können eingelernt und drohende Ausfälle verhindert werden. Das System lernte aus den generierten Daten und die automatisierte Erfolgsquote konnte signifikant gesteigert werden.

Die entwickelten Module können, je nach Anwendungsfeld und eingesetzten CPS, adaptiert und ausgetauscht werden. Sowohl die durchgeführte Proband*innenstudie als auch der Versuch zur Lernfähigkeit von CPS durch Teleoperation erheben nicht den Anspruch, ein repräsentatives Ergebnis darzustellen. Die Versuchsreihen sollen wichtige Randbedingungen der Teleoperation aufzeigen und die Funktionalität des erarbeiteten Konzepts untermauern.

Sobald Algorithmen und zugehörige Hardware in der Lage sind, das Abstraktionsvermögen, die schnelle Lernfähigkeit und das immense Modellwissen des Menschen abzubilden, wird eine Anwendung des vorgestellten Frameworks redundant. Doch bis zu diesem Zeitpunkt, kann das vorgestellte Konzept durch intuitives Einbeziehen menschlicher Fähigkeiten und automatisierter Lerndatengenerierung zur kontinuierlichen Verbesserung von CPS verwendet werden.

9 Anhang

9.1 Trainingsparameter

```
eval:
  batch_size: 1
  cache_images: false
  class_image_augmentation: ''
  dataset_names:
  - ifl-all
  dataset_scales:
  - 1745.0
  iter: 5000
  mAP_iou_thresholds:
  - 0.5
  nms_across_classes: false
  nms_iou_threshold: 0.3
  nms_score_threshold: -inf
  scales_of_image_pyramid:
  - 0.5, 0.625, 0.8, 1, 1.2, 1.4, 1.6
  train_subset_for_eval_size: 0
init:
  model: models/os2d_v2-init.pth
  transform: models/weakalign_resnet101_affine_tps.pth.tar
is_cuda: true
model:
  backbone_arch: ResNet50
  class_image_size: 120
  merge_branch_parameters: true
  normalization_mean:
  - 0.485, 0.456, 0.406
  normalization_std:
  - 0.229, 0.224, 0.225
  use_group_norm: false
  use_inverse_geom_model: true
  use_simplified_affine_model: false
output:
  best_model:
```

```
dataset: ''
do_get_best_model: true
metric: mAP@0.50
mode: max
path: output/ifl
print_iter: 1
save_iter: 10000
save_log_to_file: false
random_seed: 0
train:
  augment:
    jitter_aspect_ratio: 0.9
    min_box_coverage: 0.7
    mine_extra_class_images: false
    random_color_distortion: false
    random_crop_class_images: false
    random_flip_batches: true
    scale_jitter: 0.7
    train_patch_height: 300
    train_patch_width: 300
  batch_size: 6
  cache_images: false
  class_batch_size: 30
  dataset_name: ifl-all
  dataset_scale: 1745.0
  do_training: true
  mining:
    do_mining: false
  model:
    freeze_bn: true
    freeze_bn_transform: true
    freeze_transform: false
    num_frozen_extractor_blocks: 0
    train_features: true
    train_transform_on_negs: false
  objective:
    class_neg_weight: 1.0
    class_objective: RLL
    loc_weight: 0.0
    neg_margin: 0.5
    neg_to_pos_ratio: 3
    negative_iou_threshold: 0.1
    pos_margin: 0.6
    positive_iou_threshold: 0.5
    remap_classification_targets: true
    remap_classification_targets_iou_neg: 0.4
    remap_classification_targets_iou_pos: 0.8
```

```
rll_neg_weight_ratio: 0.001
optim:
  anneal_lr:
    cooldown: 10000
    gamma: 0.1
    initial_patience: 0
    milestones:
      - 100000, 150000
    min_value: 1.0e-05
    patience: 1000
    quantity_epsilon: 0.01
    quantity_mode: max
    quantity_smoothness: 2000
    quantity_to_monitor: mAP@0.50_ifl
    reduce_factor: 0.5
    reload_best_model_after_anneal_lr: true
    type: MultiStepLR
  lr: 0.0001
  max_grad_norm: 100.0
  max_iter: 200000
  optim_method: sgd
  sgd_momentum: 0.9
  weight_decay: 0.0001
```

9.2 Datenkompression

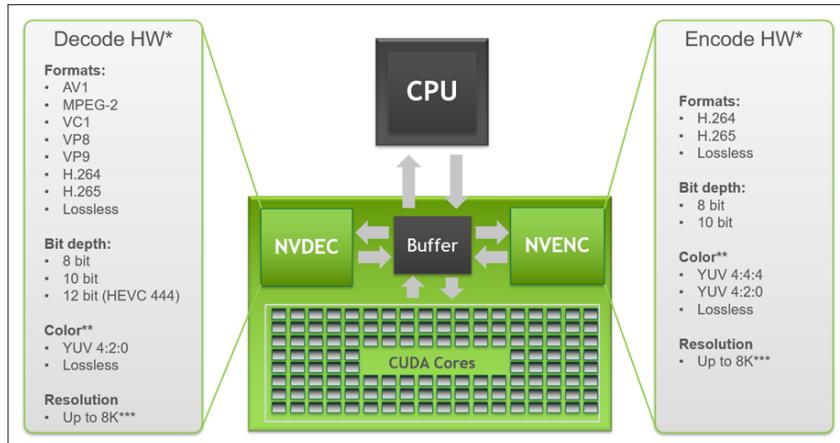


Abbildung 9.1: Struktur der hardwarebasierten Videokomprimierung auf einer Nvidia-GPU [NVIDIA(2013)]

10 Studentische Arbeiten

Während meiner Zeit am IFL und insbesondere im Rahmen des Forschungsprojekts QBIIK [BMW(2020a)] wurde meine Forschung durch etliche Seminararbeiten, Bachelor- und Masterarbeiten prägend unterstützt. Ich möchte mich an dieser Stelle ganz herzlich bei allen Studierenden bedanken, die meine Begeisterung im Bereich Teleoperation und Robotik in der Logistik geteilt und durch ihre Abschlussarbeiten in den unterschiedlichsten Bereichen zu dem Gesamtbild dieser Dissertation beigetragen haben. Herzlichen Dank! Die Arbeit mit euch hat mir stets viel Spaß gemacht.

Selina Brendle, Entwicklung einer virtuellen, dreidimensionalen Interaktionsoberfläche für Intralogistiksysteme, Bachelorarbeit, 2016 [Brendle(2016)]

Johannes Klein, Konzipierung und Entwicklung eines VR-Fernleitstandes, Bachelorarbeit, 2017 [Klein(2017)]

Johannes Kirschnick, Entwicklung einer Gimbalsteuerung für eine kamerabasierte Personenverfolgung, Bachelorarbeit, 2017 [Kirschnick(2017)]

Clemens Schröder, Bahnplanung eines 6-R-Knickarmroboters in der virtuellen Realität mittels analytischer Lösung der inversen Kinematik, Masterarbeit, 2018 [Schröder(2018)]

Constantin Enke, Bereitstellung von visuellen Informationen in einer VR-Umgebung zur Fernsteuerung eines Knickarmroboters, Bachelorarbeit, 2018 [Enke(2018)]

Lukas Hägele, Reinforcement Learning basierte Verfahren zur Repräsentation der Inversen Kinematik eines Knickarmroboters, Masterarbeit, 2018 [Hägele(2018)]

Benedikt Mayer, Immersive Visualization of the Environment using Virtual Reality and a Gimbal, Masterarbeit, 2019 [Mayer(2019)]

Mengxing Hu, Closing the Gap of Human and Robot Grasping with the help of VR-Gloves, Masterarbeit, 2019 [Mengxing(2019)]

Ziquing Zhao, Imitation and Reinforcement Learning for trajectory planning of 7-DoF Robotic Arm, Masterarbeit, 2019 [Zhao(2019)]

Yi Wen, Anomaly detection of robotic grasping process, Masterarbeit, 2019 [Wen(2019)]

Philipp Altoe, Inverse Kinematik für Gelenkarmroboter mit 7 Freiheitsgraden unter Berücksichtigung verschiedener Kriterien zur Auflösung der Redundanz, Masterarbeit, 2019 [Altoé(2019)]

Bernhard Knievel, Generation of visual learning data from the autonomous operation and teleoperation of a logistics robot, Masterarbeit, 2020 [Knievel(2020)]

Constantin Enke, Adaptives Ausführen logistischer Aufgaben durch einen Roboter auf Basis vorgemachter Trajektorien, Masterarbeit, 2021 [Enke(2021)]

Tabellenverzeichnis

2.1	Die 10 Stufen der Autonomie, adaptiert aus Sheridan [Sheridan und Verplank(1978)]	17
3.1	Die vier Schichten des TCP/IP-Referenzmodell, sowie mit (*) markierte Einträge aus dem hybriden Referenzmodell, adaptiert aus [Baun(2020)]	36
3.2	Datenübertragungsraten für diverse IEEE-Standards für WLAN, adaptiert aus [Baun(2020)]	38
3.3	MTU für unterschiedliche Netzwerktechnologien, adaptiert aus [Baun(2020)]	39
5.1	Übersicht über die Datenmenge von Farbbild, Tiefenbild und Pointcloud bei einer Aktualisierungsrate von 30 Hz . (1) Farbbild in HD Auflösung (2) Farbbild projiziert in Tiefenbildkoordinaten	86
5.2	Kompressionsfaktor von Farb- und Tiefenbild sowie der kolorierten Punktwolke. (1) Die angegebenen Werte beziehen sich auf die durchschnittliche Datenmenge während einer Teleoperation. (2) Die kolorierte Punktwolke wird aus komprimierten Farb- und Tiefenbild berechnet, die komprimierte Größe entspricht somit der Summe der komprimierten Bilddaten.	89
6.1	Vergleich von Trainingsdauer \bar{t} , durchschnittliche Genauigkeit \bar{a} und FLOPS pro Inferenz f in Millionen.	102
7.1	Tabellarische Übersicht über den erhobenen Datensatz bei der Untersuchung des Einflusses von Latenz auf die Teleoperation der Stufe 4 eines Kommissionierroboters. Die Erfahrungsbewertung wird in Prozent mit Bezug zur maximalen Punktzahl der Umfrage angegeben.	120
		163

7.2	Beschreibung der Visualisierung und vorgegebenen Latenz bei Versuch 1 bis Versuch 10. (*): Die Verzögerung basiert auf der ermittelten Systemlatenz in Kombination mit einer zusätzlichen künstlich vorgegebenen Netzwerklatenz bei den Versuchen V3-V10.	122
7.3	Ermittelte Teleoperationszeiten aller Proband*innen und Versuche. Versuch 1 und Versuch 2 wurden mit jeweils sechs Iterationen durchgeführt, Versuch 3 bis Versuch 10 mit jeweils zwei Iterationen. Durchschnitt (avg) und Standardabweichung (std) beziehen sich auf alle durchgeführten Iterationen je Versuch. Die Steigerungen sind prozentual zum vorherigen Versuch angegeben. Fehlversuche sind prozentual zur Anzahl an durchgeführten Teleoperationen je Versuch angeben. Alle Latenzen sind in Millisekunden [ms] und alle Versuchszeiten sind in Sekunden [s] angegeben.	123
7.4	Vergleich der durchschnittlichen und durchschnittlich minimalen Versuchszeit der Proband*innen bei V1 (keine VR-Brille) und V2 (mit VR-Brille). Zusätzlich wurden die 10 schnellsten und 10 langsamsten Proband*innen gesondert aufgelistet. Die Steigerung wird prozentual zwischen V2 und V1 betrachtet. Alle weiteren Angaben sind in Sekunden.	131
7.5	Vergleich der durchschnittlichen Teleoperationsdauer je Iteration für alle Proband*innen bei V1 und V2. Alle Angaben sind in Sekunden [s].	132
7.6	Prozentuale Verschlechterung der Teleoperationsdauer mit Bezug zu V1 (keine VR-Brille) unter Berücksichtigung der Lernfähigkeit der Proband*innen in beschriebener Gruppeneinteilung	135
7.7	Anwendung der vier Kriterien (siehe 7.2.4) auf die Regressionsergebnisse	140
7.8	Auswertung der ersten 150 Pick-and-Place Aufgaben vor einem Training der Objekterkennung	147
7.9	Auswertung der 750 Pick-and-Place Aufgaben nach einem Training der Objekterkennung und Vergleich zur vorherigen Leistungsfähigkeit	150

Abbildungsverzeichnis

1.1	Ist-Werte und Prognose(*) über die weltweiten Absätze von professionellen Service-Robotern [IFR(2020)]	1
1.2	Forschungsprojekt QBIIK, im Rahmen des PAiCE Technologieprogramms des Bundesministerium für Wirtschaft und Energie (BMWi). Autonome Logistikplattform mit intuitiver Mensch-Maschine-Schnittstelle, eigene Darstellung	2
1.3	Aufbau der vorliegenden Dissertation, eigene Darstellung	3
2.1	Prinzip und Beispiel der Person-zu-Ware Kommissionierung . .	7
2.2	Prinzip und Beispiel der Ware-zu-Person Kommissionierung . .	8
2.3	Sonderfall der Ware-zu-Person Kommissionierung. Die mobilen Roboter transportieren ganze Regale zu verschiedenen Kommissionierstationen, bei denen die gewünschten Artikel händisch entnommen werden	9
2.4	Beispiele der Ware-zu-Roboter Kommissionierung	10
2.5	Beispiele der Roboter-zu-Ware Kommissionierung	11
2.6	Intelligente Assistenten der menschlichen Kommissionierung . .	12
2.7	Grad der Automatisierung für die 18 identifizierten Teilbereiche der menschlichen Leistungsfähigkeit [McKinsey&Company(2017)]	13
2.8	Klassifikation der Arbeit nach Sheridan [Sheridan und Verplank(1978)]	15
2.9	Unterschied zwischen <i>Direkter Steuerung</i> und <i>Überwachender Steuerung</i> bei der Teleoperation nach Sheridan [Sheridan und Verplank(1978)]	16
2.10	Angenommener Verlauf der Leistungsverschlechterung bzw. des Telepräsenzerlebens in Abhängigkeit von der Zeitverzögerung (visuell bzw. visuell/haptisch). Modell basiert auf der Auswertung mehrerer Studien zum Thema Teleoperation unter Latenz [Pongrac(2008)]	18

2.11	VR-Teleoperation eines Knickarmroboters für freie Bewegungen (Greifen des silbernen Griffes) oder Kontaktaufgaben (Öffnen eines Bajonett-Verschlusses) [Pongrac(2008)]	19
2.12	Ergebnisse von Pongrac [Pongrac(2008)] für das teleoperierte Greifen (freie Teleoperation) unter Latenz	20
2.13	VR-Teleoperation eines Knickarmroboters mit vorgemachten Trajektorien zur Nachahmung durch den Roboter mit Hilfe eines Neuronalen Netzes [Zhang et al.(2018)Zhang, McCarthy, Jow, Lee, Chen, Goldberg, und Abbeel]	21
2.14	Übersicht der forschungsleitenden Fragestellungen, eigene Darstellung	24
3.1	Vergleich von kartesischen Koordinaten und Kugelkoordinaten, Abbildung adaptiert von [Wikimedia(2020a)]	27
3.2	Beispiel über Koordinatensystemdefinition an einem Knickarmroboter mit vier Freiheitsgraden nach DH-Konvention [Lynch und Park(2017)]	34
3.3	Datenkapselung durch die verschiedenen Netzwerkschichten des TCP/IP (ggf. hybriden) Referenzmodells nach [Baun(2020)]	37
3.4	Unterschied zwischen Ethernet und W-LAN Frame, Quelle: [Baun(2020)]	40
3.5	UDP-Header inklusive Pseudo-Header, [Baun(2020)]	41
3.6	TCP-Header inklusive Pseudo-Header, [Baun(2020)]	42
3.7	Schematische Darstellung der Datenübertragung via TCP, adaptiert aus [Baun(2020)]	42
3.8	Beispiel eines Binärbaums für die Huffman-Kodierung für das Wort TELEOPERATION, eigene Darstellung	44
3.9	Beispielhafte Quantisierung eines analogen Signals (links) und des RGB-Farbraums eines Bildes (rechts). In beiden Fällen ist der Effekt der Quantisierung deutlich erkennbar, eigene Darstellung	46
3.10	Darstellung der YUV-Komponenten eines Farbbildes, eigene Darstellung	47
3.11	YUV-Pixelformate, adaptiert aus [Wikimedia(2021)]	47

3.12 Ausnutzung zeitlicher Korrelationen (a) durch I-Frame (rot), B-Frame (grün) und P-Frame(blau). Die Verwendung von mehreren Referenzen (b) ist in modernen Videokompressionsalgorithmen möglich, eigene Darstellung	49
3.13 Laserscanner überwacht den Greifprozess des IFL PiRo Roboters [Colling et al.(2017)Colling, Dziedzitz, Hopfgarten, und Markert et al.]	51
3.14 Zusammenhang von Weltkoordinaten und Bildkoordinaten	53
3.15 Das Perzeptron als grundlegendes Verarbeitungselement von Künstlichen Neuronalen Netzen (KNN). Die Eingänge x_j , $j = 1, \dots, d$ werden über die Gewichte w_j zur Ausgabe y verbunden. [Alpaydin(2019)]	56
3.16 Neuronen mit Übertragungsfunktion und Aktivierungsfunktion φ [Wikimedia(2020b)]	57
3.17 Schematische Darstellung eines Mehrschicht-Perzeptron-Netzwerks mit einer Eingangsschicht, mehreren versteckten Schichten und einer Ausgangsschicht, eigene Darstellung	58
3.18 Vergleich der am häufigsten verwendeten Aktivierungsfunktionen, eigene Darstellung	58
4.1 Konzept für Teleoperation und kontinuierliches Lernen, eigene Darstellung	63
4.2 Verbindung von Mensch und Maschine durch Peripherie, eigene Darstellung	67
4.3 Die Stufen der Teleoperation in Abhängigkeit der eingesetzten Peripherie und Netzwerk-Kommunikation sowie der benötigten Autonomie des CPS, eigene Darstellung	69
5.1 Entwickelter Demonstrator mit verbauter Sensorik und Aktorik, eigene Darstellung	72
5.2 Schematischer Ablauf der einzelnen Teleoperationsmodule zwischen Teleoperator und Robotiksystem, eigene Darstellung	74

5.3	Drei unterschiedliche graphische Benutzeroberflächen mit ihren Teleoperationsstufen. a) Web-App, Teleoperationsstufen 1 und 2 b) Smartphone/Tablet-App, Teleoperationsstufen 1 und 2 c) Teleoperationssoftware, Teleoperationsstufen 3 und 4, eigene Darstellung	75
5.4	Web-Applikation für Teleoperationsstufe 1 und 2. Dargestellt ist der Einlernvorgang für ein bisher unbekanntes Objekt. Der Teleoperator kann direkt im Browser eine Boundingbox um das zu greifende Objekt einzeichnen. Eine Hilfestellung der Objekterkennung sieht analog dazu aus, eigene Darstellung	76
5.5	Darstellung von Teleoperationsstufe 1 und 2 am Beispiel der entwickelten Web-Applikation. Dargestellt ist die Unterstützung beim Greifprozess des Roboters. Der Teleoperator kann direkt im Browser die gewünschten Greifpunkte einzeichnen, eigene Darstellung	76
5.6	Darstellung von Teleoperationsstufe 1 und 2 am Beispiel der entwickelten Android-App. Der Teleoperator kann zwischen einer Aufgabenansicht (links), einer 3D-Darstellung (Mitte) und einer 2D-Darstellung wählen, in der über den Touchscreen des Smartphones eine Hilfestellung gegeben werden kann (rechts), eigene Darstellung	77
5.7	Teleoperation der Stufe 3/4 mit Visualisierung in VR am entwickelten Demonstrator, eigene Darstellung	78
5.8	Beispiel von 2D- und 3D-Bilddaten und kolorierter Punktwolke der Teleoperationskamera. Auf Grund der hohen Auflösung der Teleoperationskamera und einer Darstellung der Punkte durch Polygone wirkt die Punktwolke (trotz anderer Ansicht) wie ein Foto, eigene Darstellung	80
5.9	Extrinsische Kalibrierung der Teleoperationskamera. Erkannte Aruco-Marker werden in rot dargestellt, verdeckter Aruco-Marker mit gestrichelter roter Linie sowie die effektive Endeffektorpose in grün dargestellt, eigene Darstellung	81
5.10	Steuerung des Knickarmroboters mit Hilfe eines VR-Controllers. Die Pose der Hand des Teleoperators wird unmittelbar an den Endeffektor des Roboters übertragen. Die vollständige Steuerung des Roboters benötigt nur drei Knöpfe, eigene Darstellung	82

5.11	Vergleich von Kompressionsgröße und Kompressionsdauer von Farb- und Tiefenbild während des Teleoperationsvorgangs. Die Kompressionszeit beinhaltet die Übertragung des Bildes zur GPU, die Kompression auf der GPU sowie den Kopiervorgang des komprimierten Bildes zurück zur CPU, eigene Darstellung	88
5.12	Eingangs- und Ausgangs Ringbuffer als FiFo-Prinzip bei der Datenübertragung, eigene Darstellung	90
5.13	Netzwerk- und CPU-Auslastung während des Teleoperationsvorgangs. In Bereichen, die mit 1 kennzeichnet sind, findet lediglich die Übertragung der Umgebungsdaten in Form von Farb- und Tiefenbild statt. Die durchschnittlich benötigte Bandbreite für das Farbbild beträgt 0.26 MB/s sowie für das Tiefenbild 2.1 MB/s. Bereiche, die mit 2 gekennzeichnet sind, beinhalten zusätzlich die Steuerung des Knickarmroboters. Die benötigte Bandbreite für die Übertragung der gesamten Visualisierung liegt durchschnittlich bei 2.36 MB/s, eigene Darstellung	91
5.14	Digitaler Zwilling des Demonstrators mit optimierter Geometrie für ein schnelles Rendering. Darstellung ohne verbaute Sensorik. Gelenkwinkel, Zustand des Greifers sowie kolorierte Punktwolke können interaktiv und in Echtzeit visualisiert werden, eigene Darstellung	94
5.15	Struktur von Datenaufnahme, Datenkompression, Datenübertragung und Visualisierung mit <i>CUDA PCC</i> und <i>PCC Shader</i> zwischen Teleoperator und Robotiksystem, eigene Darstellung . . .	95
5.16	Vergleich der Dekompressionsdauer von Farb- und Tiefenbild sowie der Berechnung der kolorierten Punktwolke, eigene Darstellung	96
5.17	Schematische Darstellung des Franka-Control-Interface (FCI) [Emika(2021)]	97
6.1	Struktur der merkmalbasierten Objekterkennung <i>One-Stage One-Shot Object Detection</i> (OS2D) [Osokin et al.(2020)Osokin, Sumin, und Lomakin]	100
6.2	Rot: mögliche Greifbereiche; Grün: Objekte im Vordergrund; Blau: verdeckte Objekte, eigene Darstellung	101

6.3	Vergleich von <i>SegModal</i> mit unterschiedlichen Backbones bei einem bisher ungesehenen Objekt, eigene Darstellung	103
6.4	Funktionalität des <i>MaxRectsBssf</i> Algorithmus mit Anpassung für die Verwendung von Parallelgreifern, adaptiert aus [Jylänki(2010)]	104
6.5	Überblick über die Netzwerkarchitektur von DORI: Blaue Module sind von Mask R-CNN adaptiert, grüne Module wurden hinzugefügt. Rot gestrichelte Linien zeigen Gewichte, die zwischen den Stufen geteilt werden. Es sind verschiedene Möglichkeiten zur Kombination von Bild- und Merkmalinformationen denkbar (Knotenpunkt in Orange) [7]	108
6.6	Kommissionierte Objekte werden in beiden Bildern erkannt und segmentiert (links: vorher, rechts: nachher) [7]	110
7.1	Latenzmessung durch VR-Controller an einem zweiten Knickarmroboter, eigene Darstellung	113
7.2	Schematische Darstellung von Versuchsaufbau 1 mit zwei Knickarmrobotern, dem Referenzsignal (grün), der Netzwerkverzögerung inklusive Verzögerung des VR-Controllers (blau) sowie der tatsächlichen Position des teleoperierten Roboters (rot) und dem Stellsignal $f(t)$ an Roboter 1, eigene Darstellung	114
7.3	Veranschaulichung der Roboterbewegung in Relation zu den Eingangsdaten des Tracking-Devices bei einer teleoperierten Artikelkommissionierung. $v_{max} = 0.62 \text{ m/s}$, $a_{max} = 19.95 \text{ m/s}^2$. Die relevanten Frequenzen befinden sich zwischen 0.0 Hz und 0.5 Hz, eigene Darstellung	115
7.4	Darstellung des Testsignals $f(t)$ mit Periodendauer $p = 10$, eigene Darstellung	116
7.5	Frequenzanalyse zwischen Roboter und Referenzsignal, eigene Darstellung	117
7.6	Gemessene Latenzen bei einer künstlichen Netzwerkverzögerung von 100 ms mit dem Testsignal $f(t)$ und $p = 10$, eigene Darstellung	118
7.7	Linearer Zusammenhang zwischen Systemlatenz und künstlich erhöhter Netzwerklatenz, eigene Darstellung	119
7.8	Aufgabe der Proband*innen, die durch Teleoperation (Stufe 4) durchzuführen ist, eigene Darstellung	121

7.9	Einfluss von Latenz und Visualisierung in VR auf die benötigte Teleoperationszeit. Basierend auf der durchschnittlichen, minimalen Versuchszeit mit zugehöriger Standardabweichung aller Proband*innen und Versuche, eigene Darstellung	124
7.10	Darstellung des schnellsten Versuchs mit Robotergeschwindigkeit, teleoperierter Trajektorie und Versuchsdauer in Abhängigkeit von Visualisierung und Latenz bei einer Teleoperation der Stufe 4, eigene Darstellung a) ohne VR-Brille, 100 ms Verzögerung, Dauer: 5.2 s b) mit VR-Brille, 100 ms Verzögerung, Dauer: 5.8 s c) mit VR-Brille, 250 ms Verzögerung, Dauer: 8.1 s d) mit VR-Brille, 600 ms Verzögerung, Dauer: 29.4 s	126
7.11	Relevante Einflussfaktoren auf die Teleoperationszeit der Proband*innen. Je steiler die grüne Kurve, desto höher der Einfluss, eigene Darstellung	127
7.12	Selbsteinschätzung der Proband*innen in den jeweiligen Gruppen, eigene Darstellung	129
7.13	Lernfähigkeit aller Proband*innen, ohne VR-Brille entspricht der direkten Sicht auf den Roboter, mit VR-Brille entspricht der Sicht durch VR-Brille auf 3D-Daten. Schattierte Bereiche entsprechen der jeweiligen Standardabweichung, eigene Darstellung	133
7.14	Lernfähigkeit der einzelnen Gruppen, ohne VR-Brille entspricht der direkten Sicht auf den Roboter, mit VR-Brille entspricht der Sicht durch VR-Brille auf 3D-Daten, eigene Darstellung	134
7.15	Einfluss der Latenz auf drei unterschiedliche Gruppen der Proband*innen, eigene Darstellung	136
7.16	Ergebnis der polynomialen Regressionsoptimierung der prozentualen Verschlechterung der Teleoperationsdauer in Abhängigkeit der wirkenden Latenz. (a) 1 Segment (b) zwei Segmente (c) drei Segmente, eigene Darstellung	139
7.17	Vergleich der Regressionsergebnisse von Pongrac [Pongrac(2008)] (adaptiert auf prozentuelle Verschlechterung) und der im Rahmen dieser Arbeit ermittelten Regressionskurve, eigene Darstellung .	142
7.18	Eingelernte Referenzbilder der zu greifenden Objekte, eigene Darstellung	144

7.19 Anwendung des in Kapitel 4 vorgestellten Konzepts für kontinuierliches Lernen durch Teleoperation mit einem Kommissionierroboter, eigene Darstellung	146
7.20 Ergebnis der ersten 150 Pick-and-Place Aufgaben mit fünf unbekannten Objekten, eigene Darstellung	148
7.21 Durch Teleoperation hinzugefügte Referenzbilder während der ersten 150 Kommissionierungen auf Grund von Erkennungsfehlern der Objekterkennung, eigene Darstellung	149
7.22 Visualisierung der generierten Lerndaten nach einer erfolgreich ausgeführten Kommissionierung. Diese wurde entweder vollständig automatisiert oder mit Hilfe eines Teleoperators durchgeführt. Für das Training wurden 150 Bildpaare pro Kamera aufgenommen und automatisiert annotiert.	150
7.23 Leistungsanalyse von allen durchgeführten Pick-and-Place Aufgaben vor und nach dem Training. Nach dem Training sind lediglich zwei Eingriffe des Teleoperators auf Grund der Objekterkennung notwendig (rote Pfeile), eigene Darstellung	151
7.24 Benötigte Hilfestellung nach Trainings der Objekterkennung. Das zu erkennende Objekt hat sich während des Greifprozesses gewendet. (a): Referenzbild, (b): Hilfestellung 1, (c): Hilfestellung 2, eigene Darstellung	152
9.1 Struktur der hardwarebasierten Videokomprimierung auf einer Nvidia-GPU [NVIDIA(2013)]	160

Literaturverzeichnis

- [accenture und Welt(2020)] accenture und Die Welt. Weltmarktführer von morgen. Bericht, 2020. Link: <https://www.accenture.com/content/dam/accenture/final/markets/europe/document/Top500-Studie-Deutschland-Weltmarktf%C3%BChrer-von-morgen-2020.pdf>. [Zugegriffen am: 20.04.2024].
- [Alakuijala et al.(2019)] Alakuijala, Asseldonk, und Boukortt et al.] Jyrki Alakuijala, Ruud van Asseldonk, und Sami Boukortt et al. JPEG XL next-generation image compression architecture and coding tools. 2019. Link: <https://www.spiedigitallibrary.org/conference-proceedings-of-spie/11137/111370K/JPEG-XL-next-generation-image-compression-architecture-and-coding-tools/10.1117/12.2529237.full>. [Zugegriffen am: 14.04.2021].
- [Alpaydin(2019)] Ethem Alpaydin. *Maschinelles Lernen*. De Gruyter Oldenbourg, Berlin, Boston, May 2019. ISBN 978-3-11-061789-4. doi: 10.1515/9783110617894. Link: <https://www.degruyter.com/view/title/541594>.
- [Altoé(2019)] Philipp Altoé. Inverse Kinematik für Gelenkarmroboter mit 7 Freiheitsgraden unter Berücksichtigung verschiedener Kriterien zur Auflösung der Redundanz, Masterarbeit. Karlsruher Institut für Technologie (KIT), Institut für Fördertechnik und Logistiksysteme (IFL), 2019. Betreut von Wolfgang Wiedmeyer und Jonathan Auberle.
- [Angelov et al.(2020)] Angelov, Petkov, Shipkovenski, und Kalushkov] Vladislav Angelov, Emiliyan Petkov, Georgi Shipkovenski, und Teodor Kalushkov. Modern Virtual Reality Headsets. In *2020 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)*, pages 1–5, June 2020. doi: 10.1109/HORA49412.2020.9152604.
- [Arnold und Furmans(2019)] Dieter Arnold und Kai Furmans. *Materialfluss in Logistiksystemen*. Springer Berlin Heidelberg, Berlin, Heidelberg,

2019. ISBN 978-3-662-60387-1 978-3-662-60388-8. doi: 10.1007/978-3-662-60388-8. Link: <http://link.springer.com/10.1007/978-3-662-60388-8>. [Zugegriffen am: 16.11.2020].
- [AUDI(2022)] AUDI. Audi Sport GmbH, Audi AG Werk Böllinger Höfe, Heilbronn, Deutschland, 2022. Link: <https://www.audi-mediacenter.com:443/de/audi-sport-gmbh-ehemals-quattro-gmbh-2946>. [Zugegriffen am: 30.10.2022].
- [Baun(2020)] Christian Baun. *Computernetze kompakt: Eine an der Praxis orientierte Einführung für Studium und Berufspraxis*. IT kompakt. Springer Berlin Heidelberg, Berlin, Heidelberg, 2020. ISBN 978-3-662-59896-2 978-3-662-59897-9. doi: 10.1007/978-3-662-59897-9. Link: <http://link.springer.com/10.1007/978-3-662-59897-9>. [Zugegriffen am: 02.09.2020].
- [Beer et al.(2014)] Beer, Fisk, und Rogers] Jenay M. Beer, Arthur D. Fisk, und Wendy A. Rogers. Toward a framework for levels of robot autonomy in human-robot interaction. *Journal of human-robot interaction*, 3(2):74–99, July 2014. ISSN 2163-0364. doi: 10.5898/JHRI.3.2.Beer. Link: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5656240/>. [Zugegriffen am: 17.08.2020].
- [Bengio et al.(2016)] Bengio, Lee, Bornschein, Mesnard, und Lin] Yoshua Bengio, Dong-Hyun Lee, Jorg Bornschein, Thomas Mesnard, und Zhouhan Lin. Towards Biologically Plausible Deep Learning. *arXiv:1502.04156 [cs]*, August 2016. Link: <http://arxiv.org/abs/1502.04156>. [Zugegriffen am: 29.11.2020]. arXiv: 1502.04156.
- [Berry et al.(2020)] Berry, Mohamed, und Yap] Michael W. Berry, Azlinah Mohamed, und Bee Wah Yap, editors. *Supervised and Unsupervised Learning for Data Science*. Unsupervised and Semi-Supervised Learning. Springer International Publishing, Cham, 2020. ISBN 978-3-030-22474-5 978-3-030-22475-2. doi: 10.1007/978-3-030-22475-2. Link: <http://link.springer.com/10.1007/978-3-030-22475-2>. [Zugegriffen am: 27.11.2020].
- [BMWi(2020a)] BMWi. PAiCE Monitor Servicerobotik 2020. page 33, 2020a. Link: https://www.digitale-technologien.de/DT/Redaktion/DE/Downloads/Publikation/PAiCE_Monitor_Servicerobotik.pdf?__blob=publicationFile&v=1. [Zugegriffen am: 20.04.2024].

- [BMWi(2020b)] PAiCE BMWi. Forschungsprojekt QBIK, Förderkennzeichen BMWi01MA17005A, 2020b. Link: https://www.digitale-technologien.de/DT/Redaktion/DE/Standardartikel/PAICEProjekte/paice-projekt_qbik.html. [Zugegriffen am: 26.04.2021].
- [Bochkovskiy et al.(2020)] Bochkovskiy, Wang, und Liao] Alexey Bochkovskiy, Chien-Yao Wang, und Hong-Yuan Mark Liao. YOLOv4: Optimal Speed and Accuracy of Object Detection. *arXiv:2004.10934 [cs, eess]*, April 2020. Link: <http://arxiv.org/abs/2004.10934>. [Zugegriffen am: 30.05.2021]. arXiv: 2004.10934.
- [Brendle(2016)] Selina Brendle. Entwicklung einer virtuellen, dreidimensionalen Interaktionsoberfläche für Intralogistiksysteme, Bachelorarbeit. Karlsruher Institut für Technologie (KIT), Institut für Fördertechnik und Logistiksysteme (IFL), 2016. Betreut von Jonathan Auberle.
- [Brizzi et al.(2018)] Brizzi, Peppoloni, und Graziano et al.] Filippo Brizzi, Lorenzo Peppoloni, und Alessandro Graziano et al. Effects of Augmented Reality on the Performance of Teleoperated Industrial Assembly Tasks in a Robotic Embodiment. *IEEE Transactions on Human-Machine Systems*, 48(2):197–206, April 2018. ISSN 2168-2305. doi: 10.1109/THMS.2017.2782490. Conference Name: IEEE Transactions on Human-Machine Systems.
- [Brown(1966)] Duane C Brown. Decentering Distortion of Lenses. *PHOTOGRAMMETRIC ENGINEERING*, page 19, 1966.
- [Burger und Burge(2006)] Wilhelm Burger und Mark James Burge. *Digitale Bildverarbeitung: eine Einführung mit Java und ImageJ ; mit 16 Tabellen*. X.media.press. Springer, Berlin, 2., überarb. aufl edition, 2006. ISBN 978-3-540-30940-6. OCLC: 181518287.
- [Bär(2022)] Bär. BÄR Automation, Gemmingen, Deutschland, 2022. Link: <https://www.baer-automation.de/de/>. [Zugegriffen am: 30.10.2022].
- [Cao et al.(2019)] Cao, Preda, und Zaharia] Chao Cao, Marius Preda, und Titus Zaharia. 3D Point Cloud Compression: A Survey. pages 1–9, July 2019. ISBN 978-1-4503-6798-1. doi: 10.1145/3329714.3338130.
- [Chandola et al.(2009)] Chandola, Banerjee, und Kumar] Varun Chandola, Arindam Banerjee, und Vipin Kumar. Anomaly detection: A survey. *ACM Computing Surveys*, 41(3):1–58, July 2009. ISSN

- 0360-0300, 1557-7341. doi: 10.1145/1541880.1541882. Link: <https://dl.acm.org/doi/10.1145/1541880.1541882>. [Zugegriffen am: 30.12.2020].
- [Chandrasekaran(2014)] Siddharth Chandrasekaran. Implementing Circular Buffer in C, 2014. Link: <https://embedjournal.com/implementing-circular-buffer-embedded-c/>. [Zugegriffen am: 21.04.2021]. Section: Programming.
- [Chen et al.(2018)] Chen, Zhu, Papandreou, Schroff, und Adam] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, und Hartwig Adam. Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation. *arXiv:1802.02611 [cs]*, August 2018. Link: <http://arxiv.org/abs/1802.02611>. [Zugegriffen am: 30.05.2021]. arXiv: 1802.02611.
- [Choromanska et al.(2019)] Choromanska, Cowen, Kumaravel, und Luss et al.] Anna Choromanska, Benjamin Cowen, Sadhana Kumaravel, und Ronny Luss et al. Beyond Backprop: Online Alternating Minimization with Auxiliary Variables. *arXiv:1806.09077 [cs, stat]*, June 2019. Link: <http://arxiv.org/abs/1806.09077>. [Zugegriffen am: 29.11.2020]. arXiv: 1806.09077.
- [Cisco(2021)] Cisco. OpenH264, 2021. Link: <https://www.openh264.org/faq.html>. [Zugegriffen am: 15.04.2021].
- [Colling et al.(2017)] Colling, Dziedzitz, Hopfgarten, und Markert et al.] Dominik Colling, Jonathan Dziedzitz, Patric Hopfgarten, und Kai Markert et al. PiRo - Ein autonomes Kommissioniersystem für inhomogene, chaotische Lager. *Volume 2017*, page Issue 10, 2017. doi: 10.2195/LJ_PROC_COLLING_DE_201710_01. Link: <https://www.logistics-journal.de/proceedings/2017/4570>. [Zugegriffen am: 29.12.2020]. Publisher: Wissenschaftliche Gesellschaft für Technische Logistik.
- [Colling et al.(2018)] Colling, Dziedzitz, Furmans, Hopfgarten, und Markert] Dominik Colling, Jonathan Dziedzitz, Kai Furmans, Patric Hopfgarten, und Kai Markert. Progress in Autonomous Picking as Demonstrated by the Amazon Robotic Challenge, 2018. Link: <https://publikationen.bibliothek.kit.edu/1000127207>. [Zugegriffen am: 29.12.2020]. Conference Name: 15th International Material Handling Research Colloquium (IMHRC 2018), Savannah, GA, USA, 23.07.2018 – 26.07.2018 Pages: 7.

- [Comerge(2022)] Comerge. Comerge AG, CarryPick, 2022. Link: <https://www.comerge.net/referenzen/carrypick>. [Zugegriffen am: 29.10.2022].
- [conatsera(2019)] conatsera. conatsera/k4a_cuda, 2019. Link: https://github.com/conatsera/k4a_cuda. [Zugegriffen am: 21.04.2021].
- [Conrady(1919)] A. E. Conrady. Decentred Lens-Systems. *Monthly Notices of the Royal Astronomical Society*, 79(5):384–390, March 1919. ISSN 0035-8711. doi: 10.1093/mnras/79.5.384. Link: <https://doi.org/10.1093/mnras/79.5.384>. [Zugegriffen am: 22.04.2021].
- [Covariant(2022)] Covariant. AI Robotics for the Real World, 2022. Link: <https://covariant.ai/>. [Zugegriffen am: 30.10.2022].
- [DJSCOE, Vile - Parle (W), Mumbai und Rajesh Desai et al.(2014)]
DJSCOE, Vile - Parle (W), Mumbai und Parth Rajesh Desai et al. A Review Paper on Oculus Rift-A Virtual Reality Headset. *International Journal of Engineering Trends and Technology*, 13(4):175–179, July 2014. ISSN 22315381. doi: 10.14445/22315381/IJETT-V13P237. Link: <http://www.ijettjournal.org/archive/ijett-v13p237>. [Zugegriffen am: 10.05.2021].
- [Duan und Andrychowicz et al.(2017)] Yan Duan und Marcin Andrychowicz et al. One-Shot Imitation Learning. *arXiv:1703.07326 [cs]*, December 2017. Link: <http://arxiv.org/abs/1703.07326>. [Zugegriffen am: 11.05.2021]. arXiv: 1703.07326.
- [Emika(2021)] Franka Emika. Franka Control Interface Documentation — Franka Control Interface (FCI) documentation, 2021. Link: <https://frankaemika.github.io/docs/>. [Zugegriffen am: 08.04.2021].
- [Enke(2018)] Constantin Enke. Bereitstellung von visuellen Informationen in einer VR-Umgebung zur Fernsteuerung eines Knickarmroboters, Bachelorarbeit. Karlsruher Institut für Technologie (KIT), Institut für Fördertechnik und Logistiksysteme (IFL), 2018. Betreut von Jonathan Auberle.
- [Enke(2021)] Constantin Enke. Adaptives Ausführen logistischer Aufgaben durch einen Roboter auf Basis vorgemachter Trajektorien, Masterarbeit. Karlsruher Institut für Technologie (KIT), Institut für Fördertechnik und Logistiksysteme (IFL), 2021. Betreut von Jonathan Auberle.
- [Ertel(2013)] Wolfgang Ertel. *Grundkurs Künstliche Intelligenz: eine praxisorientierte Einführung*. Lehrbuch. Springer Vieweg, Wiesbaden, 3. aufl

- edition, 2013. ISBN 978-3-8348-2157-7 978-3-8348-1677-1. OCLC: 859547229.
- [Ertoz et al.(2002)] Ertoz, Steinbach, und Kumar] Levent Ertoz, Michael Steinbach, und Vipin Kumar. A New Shared Nearest Neighbor Clustering Algorithm and its Applications. In *Workshop on Clustering High Dimensional Data and its Applications at 2nd SIAM International Conference on Data Mining*, 2002.
- [Frohm et al.(2008)] Frohm, Lindström, und Winroth et al.] J Frohm, V Lindström, und M Winroth et al. Levels of Automation in Manufacturing. page 29, 2008.
- [Garrido-Jurado et al.(2015)] Sergio Garrido-Jurado et al. Generation of fiducial marker dictionaries using Mixed Integer Linear Programming. *Pattern Recognition*, 51, October 2015. doi: 10.1016/j.patcog.2015.09.023.
- [Gattringer(2011)] Hubert Gattringer. *Starr-elastische Robotersysteme*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011. ISBN 978-3-642-22827-8 978-3-642-22828-5. doi: 10.1007/978-3-642-22828-5. Link: <http://link.springer.com/10.1007/978-3-642-22828-5>. [Zugegriffen am: 25.08.2020].
- [Geiger et al.(2010)] Geiger, Popp, Färber, Artigas, und Kremer] Lena Geiger, Michael Popp, Berthold Färber, Jordi Artigas, und Philipp Kremer. The Influence of Telemanipulation-Systems on Fine Motor Performance. In *2010 Third International Conference on Advances in Computer-Human Interactions*, pages 44–49, February 2010. doi: 10.1109/ACHI.2010.46.
- [Geisberger und Broy(2012)] Eva Geisberger und Manfred Broy. Cyber-Physical Systems: Visionen, Charakteristika und Neue Fähigkeiten. In Eva Geisberger und Manfred Broy, editors, *agendaCPS: Integrierte Forschungsagenda Cyber-Physical Systems*, acatech STUDIE, pages 29–68. Springer, Berlin, Heidelberg, 2012. ISBN 978-3-642-29099-2. doi: 10.1007/978-3-642-29099-2_2. Link: https://doi.org/10.1007/978-3-642-29099-2_2. [Zugegriffen am: 19.08.2023].
- [George und Floerkemeier(2014)] Marian George und Christian Floerkemeier. Recognizing Products: A Per-exemplar Multi-label Image Classification Approach. In David Fleet, Tomas Pajdla, Bernt Schiele, und Tinne Tuytelaars, editors, *Computer Vision – ECCV 2014*, volume 8690, pages 440–455. Springer International Publishing, Cham, 2014. ISBN 978-3-319-10604-5 978-3-319-10605-2. doi: 10.1007/978-3-319-10605-

- 2_29. Link: http://link.springer.com/10.1007/978-3-319-10605-2_29. [Zugegriffen am: 20.06.2021]. Series Title: Lecture Notes in Computer Science.
- [Gerschütz et al.(2019)] Gerschütz, Fechter, Schleich, und Wartzack
Benjamin Gerschütz, Marius Fechter, Benjamin Schleich, und Sandro Wartzack. A Review of Requirements and Approaches for Realistic Visual Perception in Virtual Reality. *Proceedings of the Design Society: International Conference on Engineering Design*, 1(1):1893–1902, July 2019. ISSN 2220-4342. doi: 10.1017/dsi.2019.195. Link: <https://www.cambridge.org/core/journals/proceedings-of-the-international-conference-on-engineering-design/article/review-of-requirements-and-approaches-for-realistic-visual-perception-in-virtual-reality/7FBBB5E017B67176B13560FB193781F2>. [Zugegriffen am: 10.05.2021]. Publisher: Cambridge University Press.
- [Google(2021)] Google. FlatBuffers: FlatBuffers white paper, 2021. Link: https://google.github.io/flatbuffers/flatbuffers_white_paper.html. [Zugegriffen am: 06.07.2021].
- [GreyOrange(2021)] GreyOrange. Ranger GTP, 2021. Link: <https://www.greyorange.com/de/fos-overview/ranger-roboterserie/ranger-gtp/>. [Zugegriffen am: 07.05.2021].
- [Gruen et al.(2020)] Gruen, Ofek, Steed, Gal, Sinclair, und Gonzalez-Franco
Robert Gruen, Eyal Ofek, Anthony Steed, Ran Gal, Mike Sinclair, und Mar Gonzalez-Franco. Measuring System Visual Latency through Cognitive Latency on Video See-Through AR devices. page 9, 2020.
- [Haykin(2009)] Simon S. Haykin. *Neural networks and learning machines*. Prentice Hall, New York, 3rd ed edition, 2009. ISBN 978-0-13-147139-9. Link: <https://dai.fmph.uniba.sk/courses/NN/haykin.neural-networks.3ed.2009.pdf>. [Zugegriffen am: 20.04.2024]. OCLC: ocn237325326.
- [He et al.(2015)] He, Zhang, Ren, und Sun] Kaiming He, Xiangyu Zhang, Shaoqing Ren, und Jian Sun. Deep Residual Learning for Image Recognition. *arXiv:1512.03385 [cs]*, December 2015. Link: <http://arxiv.org/abs/1512.03385>. [Zugegriffen am: 30.05.2021]. arXiv: 1512.03385.

- [He et al.(2018)] He, Gkioxari, Dollár, und Girshick] Kaiming He, Georgia Gkioxari, Piotr Dollár, und Ross Girshick. Mask R-CNN. *arXiv:1703.06870 [cs]*, January 2018. Link: <http://arxiv.org/abs/1703.06870>. [Zugegriffen am: 04.06.2021]. arXiv: 1703.06870.
- [Hopfgarten et al.(2020)] Hopfgarten, Auberle, und Hein] P. Hopfgarten, J. Auberle, und B. Hein. Grasp Area Detection of Unknown Objects based on Deep Semantic Segmentation. In *2020 IEEE 16th International Conference on Automation Science and Engineering (CASE)*, pages 804–809, August 2020. doi: 10.1109/CASE48305.2020.9217014. ISSN: 2161-8089.
- [Hornung et al.(2013)] Hornung, Wurm, Bennewitz, Stachniss, und Burgard] Armin Hornung, Kai M. Wurm, Maren Bennewitz, Cyrill Stachniss, und Wolfram Burgard. OctoMap: an efficient probabilistic 3D mapping framework based on octrees. *Autonomous Robots*, 34(3):189–206, April 2013. ISSN 1573-7527. doi: 10.1007/s10514-012-9321-0. Link: <https://doi.org/10.1007/s10514-012-9321-0>. [Zugegriffen am: 09.04.2021].
- [Huang(2017)] L. Huang. *A Concise Introduction to Mechanics of Rigid Bodies*. Springer International Publishing, Cham, 2017. ISBN 978-3-319-45040-7 978-3-319-45041-4. doi: 10.1007/978-3-319-45041-4. Link: <http://link.springer.com/10.1007/978-3-319-45041-4>. [Zugegriffen am: 21.08.2020].
- [Huffman(1952)] David A. Huffman. A method for the construction of minimum-redundancy codes. *Proceedings of the IRE*, 40(9):1098–1101, 1952. doi: 10.1109/JRPROC.1952.273898.
- [Hägele(2018)] Lukas Hägele. Reinforcement Learning basierte Verfahren zur Repräsentation der Inversen Kinematik eines Knickarmroboters, Masterarbeit. Karlsruher Institut für Technologie (KIT), Institut für Fördertechnik und Logistiksysteme (IFL), 2018. Betreut von Jonathan Auberle.
- [IAM(2021)] IAM. IAM Robotics: Swift Robot | Autonomous Material Picking Robot with Obstacle Detection, 2021. Link: <https://www.iamrobotics.com/products/swift/>. [Zugegriffen am: 07.05.2021].
- [IDS(2022a)] IDS. 3D-Kamera Ensenso N-Serie - IDS Imaging Development Systems GmbH, 2022a. Link: <https://de.ids-imaging.com/ensenso-3d-camera-n-series.html>. [Zugegriffen am: 30.10.2022].

- [IDS(2022b)] IDS. UI-5270CP Rev. 2, 2022b. Link: <https://de.ids-imaging.com/store/ui-5270cp-rev-2.html>. [Zugegriffen am: 30.10.2022].
- [IFL-CAMP(2021)] IFL-CAMP. IFL-CAMP/easy_handeye, April 2021. Link: https://github.com/IFL-CAMP/easy_handeye. [Zugegriffen am: 09.04.2021].
- [IFR(2019)] IFR. International Federation of Robotics: World Robotics Presentation, IFR Press Conference, 2019. Link: <https://ifr.org/downloads/press2018/IFR%20World%20Robotics%20Presentation%20-%2018%20Sept%202019.pdf>. [Zugegriffen am: 19.08.2020].
- [IFR(2020)] IFR. International Federation of Robotics: World Robotics Presentation, IFR Press Conference, 2020. Link: https://ifr.org/downloads/press2018/Presentation_WR_2020.pdf. [Zugegriffen am: 27.06.2021].
- [Ilg und Groß(2017)] Uwe Ilg und Julia Groß. Sehen & Sehbahn: von der Netzhaut ins Gehirn, 2017. Link: <https://www.dasgehirn.info/wahrnehmen/sehen/die-sehbahn-hochgeschwindigkeitsleitung-ins-gehirn>. [Zugegriffen am: 13.05.2021].
- [ISO(2021)] ISO. ISO/IEC 23090-5 - Coded representation of immersive media — Visual Volumetric Video-based Coding (V3C) and Video-based Point Cloud Compression (V-PCC), 2021. Link: <https://www.iso.org/cms/render/live/en/sites/iso.org/contents/data/standard/07/30/73025.html>. [Zugegriffen am: 13.04.2021].
- [ISO(2021)] ISO. ISO/IEC JTC 1/SC 29 - Coding of audio, picture, multimedia and hypermedia information, 2021. Link: <https://www.iso.org/cms/render/live/en/sites/iso.org/contents/data/committee/04/53/45316.html>. [Zugegriffen am: 13.04.2021].
- [Ittiyavirah et al.(2013)] Ittiyavirah, Jones, und Siddarth] Sibi Ittiyavirah, S. Jones, und P. Siddarth. Analysis of different activation functions using backpropagation neural networks. *Journal of Theoretical and Applied Information Technology*, 47:1344–1348, 01 2013.

- [Jaderberg et al.(2016)] Jaderberg, Simonyan, Zisserman, und Kavukcuoglu] Max Jaderberg, Karen Simonyan, Andrew Zisserman, und Koray Kavukcuoglu. Spatial Transformer Networks. *arXiv:1506.02025 [cs]*, February 2016. Link: <http://arxiv.org/abs/1506.02025>. [Zugegriffen am: 30.05.2021]. arXiv: 1506.02025.
- [Jaghbeer et al.(2020)] Jaghbeer, Hanson, und Johansson] Yasmeen Jaghbeer, Robin Hanson, und Mats Ingemar Johansson. Automated order picking systems and the links between design and performance: a systematic literature review. *International Journal of Production Research*, 58(15):4489–4505, August 2020. ISSN 0020-7543. doi: 10.1080/00207543.2020.1788734. Link: <https://www.tandfonline.com/doi/citedby/10.1080/00207543.2020.1788734>. [Zugegriffen am: 07.05.2021]. Publisher: Taylor & Francis.
- [Jain et al.(1995)] Jain, Kasturi, und Schunck] R. Jain, R. Kasturi, und B.G. Schunck. *Machine Vision*. Computer science series. McGraw-Hill, 1995. ISBN 978-0-07-113407-1. Link: <https://books.google.de/books?id=POSaQgAACAAJ>.
- [James et al.(2018)] James, Bloesch, und Davison] Stephen James, Michael Bloesch, und Andrew J. Davison. Task-Embedded Control Networks for Few-Shot Imitation Learning. *arXiv:1810.03237 [cs]*, October 2018. Link: <http://arxiv.org/abs/1810.03237>. [Zugegriffen am: 11.05.2021]. arXiv: 1810.03237.
- [Jylänki(2010)] Jukka Jylänki. *A thousand ways to pack the bin - a practical approach to two-dimensional rectangle bin packing*. 2010. Link: <https://raw.githubusercontent.com/rougier/freetype-gl/master/doc/RectangleBinPack.pdf>. [Zugegriffen am: 02.06.2021].
- [Kent et al.(2017)] Kent, Saldanha, und Chernova] David Kent, Carl Saldanha, und Sonia Chernova. A Comparison of Remote Robot Teleoperation Interfaces for General Object Manipulation. In *Proceedings of the 2017 ACM/IEEE International Conference on Human-Robot Interaction*, pages 371–379, Vienna Austria, March 2017. ACM. ISBN 978-1-4503-4336-7. doi: 10.1145/2909824.3020249. Link: <https://dl.acm.org/doi/10.1145/2909824.3020249>. [Zugegriffen am: 10.05.2021].
- [Kent et al.(2020)] Kent, Saldanha, und Chernova] David Kent, Carl Saldanha, und Sonia Chernova. Leveraging depth data in remote robot teleoperati-

- on interfaces for general object manipulation. *The International Journal of Robotics Research*, 39(1):39–53, January 2020. ISSN 0278-3649, 1741-3176. doi: 10.1177/0278364919888565. Link: <http://journals.sagepub.com/doi/10.1177/0278364919888565>. [Zugegriffen am: 10.05.2021].
- [Kirschnick(2017)] Johannes Kirschnick. Entwicklung einer Gimbalsteuerung für eine kamerabasierte Personenverfolgung, Bachelorarbeit. Karlsruher Institut für Technologie (KIT), Institut für Fördertechnik und Logistiksysteme (IFL), 2017. Betreut von Jonathan Auberle.
- [Klein(2017)] Johannes Klein. Konzipierung und Entwicklung eines VR-Fernleitstandes, Bachelorarbeit. Karlsruher Institut für Technologie (KIT), Institut für Fördertechnik und Logistiksysteme (IFL), 2017. Betreut von Jonathan Auberle.
- [Klerk und Fran(2017)] de Klerk und Fran. Dematic to Build Advanced Warehouse Solution for Sigma Pharmaceuticals, February 2017. Link: <https://sclaa.com.au/dematic-to-build-advanced-warehouse-solution-for-sigma-pharmaceuticals/>. [Zugegriffen am: 01.10.2020].
- [Kliuchnikov und Upenik et al.(2020)] Evgenii Kliuchnikov und Evgeniy Upenik et al. Benchmarking JPEG XL lossy/lossless image compression. In *Optics, Photonics and Digital Technologies for Imaging Applications VI*, 2020. Link: <http://infoscience.epfl.ch/record/277420/files/Submitted%20manuscript.pdf>. [Zugegriffen am: 14.04.2021].
- [Knievel(2020)] Bernhard Knievel. Generation of visual learning data from the autonomous operation and teleoperation of a logistics robot, Masterarbeit. Karlsruher Institut für Technologie (KIT), Institut für Fördertechnik und Logistiksysteme (IFL), 2020. Betreut von Jonathan Auberle.
- [Kumra und Kanan(2017)] Sulabh Kumra und Christopher Kanan. Robotic Grasp Detection using Deep Convolutional Neural Networks. *arXiv:1611.08036 [cs]*, July 2017. Link: <http://arxiv.org/abs/1611.08036>. [Zugegriffen am: 30.05.2021]. arXiv: 1611.08036.
- [Lenz et al.(2015)] Lenz, Lee, und Saxena] Ian Lenz, Honglak Lee, und Ashutosh Saxena. Deep learning for detecting robotic grasps. *The International Journal of Robotics Research*, 34(4-5):705–724, April 2015. ISSN 0278-3649. doi: 10.1177/0278364914549607. Link:

- <https://doi.org/10.1177/0278364914549607>. [Zugegriffen am: 30.05.2021]. Publisher: SAGE Publications Ltd STM.
- [libpng(2021)] libpng. libpng Home Page, 2021. Link: <http://www.libpng.org/pub/png/libpng.html>. [Zugegriffen am: 13.04.2021].
- [Lin et al.(2017)] Lin, Dollár, Girshick, He, Hariharan, und Belongie] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, und Serge Belongie. Feature Pyramid Networks for Object Detection. *arXiv:1612.03144 [cs]*, April 2017. Link: <http://arxiv.org/abs/1612.03144>. [Zugegriffen am: 04.06.2021]. arXiv: 1612.03144.
- [Lloyd(1982)] S. Lloyd. Least squares quantization in PCM. *IEEE Transactions on Information Theory*, 28(2):129–137, March 1982. ISSN 0018-9448. doi: 10.1109/TIT.1982.1056489. Link: <http://ieeexplore.ieee.org/document/1056489/>. [Zugegriffen am: 30.12.2020].
- [Long et al.(2015)] Long, Shelhamer, und Darrell] Jonathan Long, Evan Shelhamer, und Trevor Darrell. Fully Convolutional Networks for Semantic Segmentation. *arXiv:1411.4038 [cs]*, March 2015. Link: <http://arxiv.org/abs/1411.4038>. [Zugegriffen am: 30.05.2021]. arXiv: 1411.4038.
- [Lynch und Park(2017)] Kevin M. Lynch und Frank C. Park. *Modern robotics: mechanics, planning, and control*. Cambridge University Press, Cambridge, UK, 2017. ISBN 978-1-107-15630-2 978-1-316-60984-2. OCLC: ocn983881868.
- [Ma et al.(2019)] Ma, Lewis, und Kleijn] Wan-Duo Kurt Ma, J. P. Lewis, und W. Bastiaan Kleijn. The HSIC Bottleneck: Deep Learning without Back-Propagation. *arXiv:1908.01580 [cs, stat]*, December 2019. Link: <http://arxiv.org/abs/1908.01580>. [Zugegriffen am: 29.11.2020]. arXiv: 1908.01580.
- [Magazino(2022a)] Magazino. Lagerroboter von Magazino gehen in den Pilotbetrieb bei ZF, February 2022a. Link: <https://telematik-markt.de/telematik/lagerroboter-von-magazino-gehen-den-pilotbetrieb-bei-zf>. [Zugegriffen am: 30.10.2022].
- [Magazino(2022b)] Magazino. SOTO - Supply Chain-Lösung für die Produktionslogistik - Magazino, 2022b. Link: <https://www.magazino.eu/produkte/soto/>. [Zugegriffen am: 30.10.2022].

- [Manz(2020)] Olaf Manz. *Gut gepackt - Kein Bit zu viel: Kompression digitaler Daten verständlich erklärt*. essentials. Springer Fachmedien Wiesbaden, Wiesbaden, 2020. ISBN 978-3-658-31215-2 978-3-658-31216-9. doi: 10.1007/978-3-658-31216-9. Link: <http://link.springer.com/10.1007/978-3-658-31216-9>. [Zugegriffen am: 10.09.2020].
- [Mareczek(2020)] Jörg Mareczek. *Grundlagen der Roboter-Manipulatoren – Band 1: Modellbildung von Kinematik und Dynamik*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2020. ISBN 978-3-662-52758-0 978-3-662-52759-7. doi: 10.1007/978-3-662-52759-7. Link: <http://link.springer.com/10.1007/978-3-662-52759-7>. [Zugegriffen am: 24.08.2020].
- [Mayer(2019)] Benedikt Mayer. Immersive Visualization of the Environment using Virtual Reality and a Gimbal, Masterarbeit. Karlsruher Institut für Technologie (KIT), Institut für Fördertechnik und Logistiksysteme (IFL), 2019. Betreut von Jonathan Auberle.
- [McKinsey&Company(2017)] McKinsey&Company. A FUTURE THAT WORKS: AUTOMATION, EMPLOYMENT, AND PRODUCTIVITY. Bericht, 2017. Link: <https://www.handelsblatt.com/downloads/19337114/1/mgi.pdf?ticket=ST-11682397-x3Bru7mQOVt134d0qkMg-ap2>. [Zugegriffen am: 19.08.2020].
- [Mehrotra et al.(2017)] Mehrotra, Mohan, und Huang] Kishan G. Mehrotra, Chilukuri K. Mohan, und HuaMing Huang. *Anomaly Detection Principles and Algorithms*. Terrorism, Security, and Computation. Springer International Publishing, Cham, 2017. ISBN 978-3-319-67524-4 978-3-319-67526-8. doi: 10.1007/978-3-319-67526-8. Link: <http://link.springer.com/10.1007/978-3-319-67526-8>. [Zugegriffen am: 30.12.2020].
- [Mengxing(2019)] Hu Mengxing. Closing the Gap of Human and Robot Grasping with the help of VR-Gloves, Masterarbeit. Karlsruher Institut für Technologie (KIT), Institut für Fördertechnik und Logistiksysteme (IFL), 2019. Betreut von Jonathan Auberle.
- [Mhaskar und Micchelli(1993)] H. N. Mhaskar und C. A. Micchelli. How to choose an activation function. In *Proceedings of the 6th International Conference on Neural Information Processing Systems*, NIPS'93, pages 319–326, San Francisco, CA, USA, 1993. Morgan Kaufmann Publishers Inc.

- [Microsoft(2021a)] Microsoft. microsoft/Azure-Kinect-Sensor-SDK, April 2021a. Link: <https://github.com/microsoft/Azure-Kinect-Sensor-SDK>. [Zugegriffen am: 09.04.2021].
- [Microsoft(2021b)] Microsoft. microsoft/Azure_kinect_ros_driver, April 2021b. Link: https://github.com/microsoft/Azure_Kinect_ROS_Driver. [Zugegriffen am: 09.04.2021].
- [Nelder und Wedderburn(1972)] J. A. Nelder und R. W. M. Wedderburn. Generalized Linear Models. *Journal of the Royal Statistical Society. Series A (General)*, 135(3):370, 1972. ISSN 00359238. doi: 10.2307/2344614. Link: https://www.jstor.org/stable/10.2307/2344614?orig_in=crossref. [Zugegriffen am: 06.05.2021].
- [NVIDIA(2013)] NVIDIA. NVIDIA VIDEO CODEC SDK, August 2013. Link: <https://developer.nvidia.com/nvidia-video-codec-sdk>. [Zugegriffen am: 15.04.2021].
- [Oitzman(2018)] Mike Oitzman. Robotics Business Review, April 2018. Link: <https://www.robotsbusinessreview.com/supply-chain/more-robot-picks-to-see-at-modex-2018/>. [Zugegriffen am: 19.08.2023].
- [Osa et al.(2018)] Osa, Pajarinen, Neumann, Bagnell, Abbeel, und Peters] Takayuki Osa, Joni Pajarinen, Gerhard Neumann, J. Andrew Bagnell, Pieter Abbeel, und Jan Peters. An Algorithmic Perspective on Imitation Learning. *Foundations and Trends in Robotics*, 7(1-2):1–179, 2018. ISSN 1935-8253, 1935-8261. doi: 10.1561/2300000053. Link: <http://www.nowpublishers.com/article/Details/ROB-053>. [Zugegriffen am: 11.05.2021].
- [Osokin et al.(2020)] Osokin, Sumin, und Lomakin] Anton Osokin, Denis Sumin, und Vasily Lomakin. OS2D: One-Stage One-Shot Object Detection by Matching Anchor Features. *arXiv:2003.06800 [cs]*, August 2020. Link: <http://arxiv.org/abs/2003.06800>. [Zugegriffen am: 23.04.2021]. arXiv: 2003.06800.
- [Parasuraman et al.(2000)] Parasuraman, Sheridan, und Wickens] R. Parasuraman, T.B. Sheridan, und C.D. Wickens. A model for types and levels of human interaction with automation. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, 30(3):286–297, May 2000. ISSN 1558-2426. doi: 10.1109/3468.844354. Conference

- Name: IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans.
- [Peppoloni et al.(2015)] Peppoloni, Brizzi, Ruffaldi, und Avizzano] Lorenzo Peppoloni, Filippo Brizzi, Emanuele Ruffaldi, und Carlo Alberto Avizzano. Augmented reality-aided tele-presence system for robot manipulation in industrial manufacturing. In *Proceedings of the 21st ACM Symposium on Virtual Reality Software and Technology*, pages 237–240, Beijing China, November 2015. ACM. ISBN 978-1-4503-3990-2. doi: 10.1145/2821592.2821620. Link: <https://dl.acm.org/doi/10.1145/2821592.2821620>. [Zugegriffen am: 10.05.2021].
- [Plange(2017)] Niklas Plange. Sehnerv und Sehbahn. In Peter Walter und Niklas Plange, editors, *Basiswissen Augenheilkunde*, Springer-Lehrbuch, pages 291–309. Springer, Berlin, Heidelberg, 2017. ISBN 978-3-662-52801-3. doi: 10.1007/978-3-662-52801-3_20. Link: https://doi.org/10.1007/978-3-662-52801-3_20. [Zugegriffen am: 13.05.2021].
- [Pongrac(2008)] Helena Pongrac. *Gestaltung und Evaluation von virtuellen und Telepräsenzsystemen an Hand von Aufgabenleistung und Präsenzempfinden*. PhD thesis, Universität der Bundeswehr, München, 2008. Link: <https://pdfs.semanticscholar.org/0461/2cd597aa38aa318777d084dcbe3dac065cd8.pdf>. [Zugegriffen am: 19.08.2020].
- [Raaen und Kjellmo(2015)] Kjetil Raaen und Ivar Kjellmo. Measuring Latency in Virtual Reality Systems. pages 457–462, September 2015. ISBN 978-3-319-24588-1. doi: 10.1007/978-3-319-24589-8_40.
- [Rao et al.(2010)] Rao, Le, Phoka, Quigley, Sudsang, und Ng] Deepak Rao, Quoc V. Le, T. Phoka, M. Quigley, A. Sudsang, und A. Ng. Grasping novel objects with depth segmentation. *undefined*, 2010. Link: <https://paper/Grasping-novel-objects-with-depth-segmentation-Rao-Le/7de2b8a2fc264b4dedfc94dcd1dd90a47e2b620f>. [Zugegriffen am: 30.05.2021].
- [Ren et al.(2016)] Ren, He, Girshick, und Sun] Shaoqing Ren, Kaiming He, Ross Girshick, und Jian Sun. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *arXiv:1506.01497 [cs]*, January 2016. Link: <http://arxiv.org/abs/1506.01497>. [Zugegriffen am: 30.05.2021]. arXiv: 1506.01497.

- [Rill et al.(2020)] Rill, Schaeffer, und Borchsenius] Georg Rill, Thomas Schaeffer, und Fredrik Borchsenius. *Grundlagen und computergerechte Methodik der Mehrkörpersimulation: Vertieft in Matlab-Beispielen, Übungen und Anwendungen*. Springer Fachmedien Wiesbaden, Wiesbaden, 2020. ISBN 978-3-658-28911-9 978-3-658-28912-6. doi: 10.1007/978-3-658-28912-6. Link: <http://link.springer.com/10.1007/978-3-658-28912-6>. [Zugegriffen am: 24.08.2020].
- [Rocco et al.(2018)] Rocco, Arandjelović, und Sivic] Ignacio Rocco, Relja Arandjelović, und Josef Sivic. End-to-end weakly-supervised semantic alignment. *arXiv:1712.06861 [cs]*, April 2018. Link: <http://arxiv.org/abs/1712.06861>. [Zugegriffen am: 30.05.2021]. arXiv: 1712.06861.
- [Romero-Ramirez et al.(2018)] Romero-Ramirez, Muñoz-Salinas, und Medina-Carnicer] Francisco Romero-Ramirez, Rafael Muñoz-Salinas, und Rafael Medina-Carnicer. Speeded Up Detection of Squared Fiducial Markers. *Image and Vision Computing*, 76, June 2018. doi: 10.1016/j.imavis.2018.05.004.
- [Rumelhart et al.(1986)] Rumelhart, Hinton, und Williams] David E. Rumelhart, Geoffrey E. Hinton, und Ronald J. Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, October 1986. ISSN 1476-4687. doi: 10.1038/323533a0. Link: <https://www.nature.com/articles/323533a0>. [Zugegriffen am: 29.11.2020]. Number: 6088 Publisher: Nature Publishing Group.
- [Ruzanski(2006)] Evan Ruzanski. Effects of MP3 encoding on the sounds of music. *Potentials, IEEE*, 25:43–45, April 2006. doi: 10.1109/MP.2006.1649011.
- [Save und Feuerberg(2014)] Luca Save und Beatrice Feuerberg. Designing Human-Automation Interaction: a new level of Automation Taxonomy. *Proceedings HFES Europe Chapter Conference Toulouse*, page 14, 2014. Link: <https://www.hfes-europe.org/wp-content/uploads/2014/06/Save.pdf>. [Zugegriffen am: 20.04.2024].
- [Schröder(2018)] Clemens Schröder. Bahnplanung eines 6-R-Knickarmroboters in der virtuellen Realität mittels analytischer Lösung der inversen Kinematik, Masterarbeit. Karlsruher Institut für Technologie (KIT), Institut für Fördertechnik und Logistiksysteme (IFL), 2018. Betreut von Jonathan Auberle.

- [secnot(2021)] secnot. secnot/rectpack, May 2021. Link: <https://github.com/secnot/rectpack>. [Zugegriffen am: 03.06.2021].
- [SelectLine(2020)] SelectLine. Effiziente Kommissionierung: Lagerplanung, Methoden, Systeme und Kommissionierzeiten, March 2020. Link: <https://blog.selectline.de/effiziente-kommissionierung/>. [Zugegriffen am: 16.11.2020].
- [Sharma et al.(2020)] Sharma, Sharma, und Athaiya] Siddharth Sharma, Simone Sharma, und Anidhya Athaiya. Activation functions in neural networks. *International Journal of Engineering Applied Sciences and Technology*, 04:310–316, 05 2020. doi: 10.33564/IJEAST.2020.v04i12.054.
- [Sheridan(2011)] Thomas B. Sheridan. Adaptive Automation, Level of Automation, Allocation Authority, Supervisory Control, and Adaptive Control: Distinctions and Modes of Adaptation. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, 41(4):662–667, July 2011. ISSN 1558-2426. doi: 10.1109/TSMCA.2010.2093888. Conference Name: IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans.
- [Sheridan und Verplank(1978)] Thomas B. Sheridan und William L. Verplank. Human and Computer Control of Undersea Teleoperators:. Bericht, Defense Technical Information Center, Fort Belvoir, VA, July 1978. Link: <http://www.dtic.mil/docs/citations/ADA057655>. [Zugegriffen am: 22.09.2020].
- [STILL(2022)] STILL. STILL, Hamburg, Deutschland, 2022. Link: <https://www.still.de/>. [Zugegriffen am: 30.10.2022].
- [Stotko und Krumpen et al.(2019)] Patrick Stotko und Stefan Krumpen et al. A VR System for Immersive Teleoperation and Live Exploration with a Mobile Robot. *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3630–3637, November 2019. doi: 10.1109/IROS40897.2019.8968598. Link: <http://arxiv.org/abs/1908.02949>. [Zugegriffen am: 10.05.2021]. arXiv: 1908.02949.
- [Strasburger und Pöppel(2002)] Hans Strasburger und Ernst Pöppel. Visual field. Encyclopedia of Neuroscience, 3rd Ed., 2004, No. 674, January 2002.
- [StrukturAG(2021)] StrukturAG. libde265 HEVC — H.265 High Efficiency Video Coding, 2021. Link: <https://www.libde265.org/>. [Zugegriffen am: 15.04.2021].

- [Strutz(2017)] Tilo Strutz. *Bilddatenkompression: Grundlagen, Codierung, Wavelets, JPEG, MPEG, H.264, HEVC*. Springer Vieweg, Wiesbaden, 5., überarb. u. erg. aufl. 2017 edition edition, April 2017. ISBN 978-3-8348-1427-2.
- [Swisslog(2022)] Swisslog. CarryPick, 2022. Link: <https://www.swisslog.com/de-de/produkte-systeme-1%cb6sungen/automatisierte-lagertechnik-lagersysteme-lagerlogistik/automatisches-kleineiteilelager/carrypick-automatisches-lager-und-kommissionier-system>. [Zugegriffen am: 29.10.2022].
- [Takahashi(2021)] Keiji Takahashi. keiji/Pcx, April 2021. Link: <https://github.com/keiji/Pcx>. [Zugegriffen am: 21.04.2021].
- [TeamViewer(2021)] TeamViewer. Vision Picking for Logistics and Warehousing, 2021. Link: <https://www.teamviewer.com/en/frontline/xpick/>. [Zugegriffen am: 09.05.2021].
- [Technologies(2021)] Unity Technologies. Unity Real-Time Development Platform | 3D, 2D VR & AR Engine, 2021. Link: <https://unity.com/>. [Zugegriffen am: 16.07.2021].
- [Technote(2022)] Technote. Bin Picking Companies List, 2022. Link: <https://technote.ai/technology/bin-picking-companies>. [Zugegriffen am: 30.10.2022].
- [Theofilis et al.(2016)] Theofilis, Orlosky, Nagai, und Kiyokawa Konstantinos Theofilis, Jason Orlosky, Yukie Nagai, und Kiyoshi Kiyokawa. Panoramic view reconstruction for stereoscopic teleoperation of a humanoid robot. In *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*, pages 242–248, Cancun, Mexico, November 2016. IEEE. ISBN 978-1-5090-4718-5. doi: 10.1109/HUMANOIDS.2016.7803284. Link: <http://ieeexplore.ieee.org/document/7803284/>. [Zugegriffen am: 10.05.2021].
- [Trafton und Schultz et al.(2004)] J Gregory Trafton und Alan C Schultz et al. Using similar representations to improve human-robot interaction. page 37, 2004. Link: https://www.researchgate.net/publication/228617004_Using_similar_representations_to_improve_human-robot_interaction. [Zugegriffen am: 20.04.2024].

- [Tsai und Lenz(1989)] R. Y. Tsai und R. K. Lenz. A new technique for fully autonomous and efficient 3D robotics hand/eye calibration. *IEEE Transactions on Robotics and Automation*, 5(3):345–358, June 1989. ISSN 2374-958X. doi: 10.1109/70.34770. Conference Name: IEEE Transactions on Robotics and Automation.
- [Tölgessy et al.(2021)] Tölgessy, Dekan, Chovanec, und Hubinsky] Michal Tölgessy, Martin Dekan, Lubos Chovanec, und Peter Hubinsky. Evaluation of the Azure Kinect and Its Comparison to Kinect V1 and Kinect V2. *Sensors*, 21(2):413, January 2021. doi: 10.3390/s21020413. Link: <https://www.mdpi.com/1424-8220/21/2/413>. [Zugegriffen am: 13.05.2021]. Number: 2 Publisher: Multidisciplinary Digital Publishing Institute.
- [ULMA(2020)] ULMA. ULMA Handling Systems, Picking Cart system, 2020. Link: <https://www.ulmahandling.com/en/automated-intralogistics/system-picking-cart>. [Zugegriffen am: 16.11.2020].
- [Viscarious(2021)] Viscarious. Fully Automated Bin Picking Robotic System – Vicarious, 2021. Link: <https://www.vicarious.com/solutions/bin-picking/>. [Zugegriffen am: 07.05.2021].
- [Wen(2019)] Yi Wen. Anomaly detection of robotic grasping process, Masterarbeit. Karlsruher Institut für Technologie (KIT), Institut für Fördertechnik und Logistiksysteme (IFL), 2019. Betreut von Jonathan Auberle.
- [Wendzel(2018)] Steffen Wendzel. *IT-Sicherheit für TCP/IP- und IoT-Netzwerke: Grundlagen, Konzepte, Protokolle, Härtung*. Springer Fachmedien Wiesbaden, Wiesbaden, 2018. ISBN 978-3-658-22602-2 978-3-658-22603-9. doi: 10.1007/978-3-658-22603-9. Link: <http://link.springer.com/10.1007/978-3-658-22603-9>. [Zugegriffen am: 07.09.2020].
- [wes b(2020)] wes b. Latency Question · Issue #1446 · microsoft/Azure-Kinect-Sensor-SDK, 2020. Link: <https://github.com/microsoft/Azure-Kinect-Sensor-SDK/issues/1446>. [Zugegriffen am: 13.05.2021].
- [Wessling(2022)] Brianna Wessling. A decade after acquiring Kiva, Amazon unveils its first AMR, 2022. Link: <https://www.therobotreport.com/a-decade-after-acquiring->

- kiva-amazon-unveils-its-first-amr/. [Zugegriffen am: 29.10.2022].
- [WIBOND(2021)] WIBOND. PickVision Pointer - WIBOND Informati onssysteme GmbH, 2021. Link: <https://wibond.de/beleglose-kommissionierung/verfahren/pick-by-spot/pickvision-pointer.html>. [Zugegriffen am: 09.05.2021].
- [Wikimedia(2020a)] Wikimedia. Kugelkoordinaten, Bildautor: Ag2gaeh (2015), 2020a. Link: <https://commons.wikimedia.org/wiki/File:Kugelkoord-def.svg>. [Zugegriffen am: 21.08.2020].
- [Wikimedia(2020b)] Wikimedia. Künstliches Neuron, July 2020b. Link: https://de.wikipedia.org/w/index.php?title=K%C3%BCnstliches_Neuron&oldid=202215811. [Zugegriffen am: 27.11.2020]. Page Version ID: 202215811.
- [Wikimedia(2021)] Wikimedia. Chroma subsampling, May 2021. Link: https://en.wikipedia.org/w/index.php?title=Chroma_subsampling&oldid=1024398475. [Zugegriffen am: 24.05.2021]. Page Version ID: 1024398475.
- [Wittenburg et al.(2014)] Wittenburg, Richard, Zierep, und Bühler] Jens Wittenburg, Hans Albert Richard, Jürgen Zierep, und Karl Bühler. *Das Ingenieurwissen: Technische Mechanik*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2014. ISBN 978-3-642-41121-2 978-3-642-41122-9. doi: 10.1007/978-3-642-41122-9. Link: <http://link.springer.com/10.1007/978-3-642-41122-9>. [Zugegriffen am: 25.08.2020].
- [Woernle(2011)] Christoph Woernle. Grundlagen der Kinematik. In *Mehr-körpersysteme*, pages 33–96. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011. ISBN 978-3-642-15981-7 978-3-642-15982-4. doi: 10.1007/978-3-642-15982-4_3. Link: http://link.springer.com/10.1007/978-3-642-15982-4_3. [Zugegriffen am: 25.08.2020].
- [Zhang et al.(2018)] Zhang, McCarthy, Jow, Lee, Chen, Goldberg, und Abbeel] Tianhao Zhang, Zoe McCarthy, Owen Jow, Dennis Lee, Xi Chen, Ken Goldberg, und Pieter Abbeel. Deep Imitation Learning for Complex Manipulation Tasks from Virtual Reality Teleoperation. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5628–5635, May 2018. doi: 10.1109/ICRA.2018.8461249. ISSN: 2577-087X.

[Zhao(2019)] Ziqing Zhao. Imitation and Reinforcement Learning for trajectory planning of 7-DoF Robotic Arm, Masterarbeit. Karlsruher Institut für Technologie (KIT), Institut für Fördertechnik und Logistiksysteme (IFL), 2019. Betreut von Jonathan Auberle.

[Ziv und Lempel(1977)] J. Ziv und A. Lempel. A universal algorithm for sequential data compression. *IEEE Transactions on Information Theory*, 23(3):337–343, May 1977. ISSN 0018-9448. doi: 10.1109/TIT.1977.1055714. Link: <http://ieeexplore.ieee.org/document/1055714/>. [Zugegriffen am: 22.05.2021].

Eigene Veröffentlichungen

Journalartikel

- [1] W. Wiedmeyer, P. Altoé, J. Auberle, C. Ledermann, and T. Kröger. A real-time-capable closed-form multi-objective redundancy resolution scheme for seven-dof serial manipulators. *IEEE Robotics and Automation Letters*, 6(2):431–438, 2021.

Konferenzbeiträge

- [2] P. Böhmler, J. Dziedzitz, P. Hopfgarten, T. Specker, and R. Lange. Efficient and precise sensor fusion for non-linear systems with out-of-sequence measurements by example of mobile robotics. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5709–5715, 2020.
- [3] Dominik Colling, Jonathan Dziedzitz, Kai Furmans, Patric Hopfgarten, and Kai Markert. Progress in autonomous picking as demonstrated by the amazon robotic challenge. In *15th International Material Handling Research Colloquium (IMHRC), Savannah, Georgia, July 23-26, 2018*, page 7 S., 2018.
- [4] Dominik Colling, Jonathan Dziedzitz, Patric Hopfgarten, Kai Markert, Kira Neubehler, Florian Eberle, Maximilian Gilles, Matthias Jung, Alp Kocabas, and Kai Furmans. Piro : Ein autonomes kommissioniersystem für inhomogene, chaotische lager = piro : An autonomous picking system for inhomogeneous, chaotic storages. *Logistics journal / Proceedings*, 2017, 2017.
- [5] Jonathan Dziedzitz, Dennis Schüthe, P. Hopfgarten, Kai Furmans, Yongzhou Zhang, I. Mamaev, and B. Hein. Eine lernende kommissionierplattform – wie roboter mit neuen technologien vom menschen lernen. 03 2019.

- [6] P. Hopfgarten, J. Auberle, and B. Hein. Grasp Area Detection of Unknown Objects based on Deep Semantic Segmentation. In *2020 IEEE 16th International Conference on Automation Science and Engineering (CASE)*, pages 804–809, August 2020. ISSN: 2161-8089.
- [7] B. Knievel, J. Auberle, and T. Kröger. Dual-shot object re-identification from RGB-D images (DORI). rejected conference paper, 06 2021.