…

Inhalt

[1 Einleitung 1](#_Toc515099986)

[1.1 Ausgangssituation und Problemstellung 1](#_Toc515099987)

[1.2 Abgrenzung der Themenstellung 1](#_Toc515099988)

[1.3 Aufbau der Arbeit 1](#_Toc515099989)

[2 Grundlagen 2](#_Toc515099990)

[2.1 Robotik im Bereich teilautonomer technischer Assistenzsysteme 2](#_Toc515099991)

[2.2 Mustererkennung als Teilgebiet des maschinellen Sehens 5](#_Toc515099992)

[2.3 Visual Servoing 9](#_Toc515099993)

[3 Konzeption 9](#_Toc515099994)

[3.1 Manipulator mit Bildverarbeitungseinheit 10](#_Toc515099995)

[3.1.1 Anforderungen an das System 10](#_Toc515099996)

[3.1.2 Kriterien für die Auswahl der Komponenten 10](#_Toc515099997)

[3.1.3 Aufbau des Systems 10](#_Toc515099998)

[3.2 Objekt- und Texterkennung 10](#_Toc515099999)

[3.2.1 Maschinelles Detektionsverfahren mit vorgegebenem Merkmalsraum 10](#_Toc515100000)

[3.2.2 Deep Learning Ansatz zur Detektion ohne Merkmalsraumvorgabe 10](#_Toc515100001)

[3.2.3 Auslesen der Tasterbeschriftung 10](#_Toc515100002)

[3.3 Objektverfolgung 11](#_Toc515100003)

[3.3.1 Echtzeit-Objektverfolgung in Videosequenzen 11](#_Toc515100004)

[3.3.2 Merkmalsabgleich und perspektivische Transformation zwischen Einzelbildern 11](#_Toc515100005)

[3.4 Entfernungsberechnung 11](#_Toc515100006)

[3.4.1 Berechnung durch Änderung des Winkels zum Ziel 11](#_Toc515100007)

[3.4.2 Berechnung durch Änderung der Bildgröße des Ziels 11](#_Toc515100008)

[3.5 Bewegungssteuerung 11](#_Toc515100009)

[3.5.1 Wahl des Koordinatensystems 11](#_Toc515100010)

[3.5.2 Positionierung des Effektors im Raum 11](#_Toc515100011)

[3.6 Integration der Teillösungen 11](#_Toc515100012)

[3.6.1 Benutzerschnittstelle 11](#_Toc515100013)

[3.6.2 Interaktion der Komponenten 11](#_Toc515100014)

[3.6.3 Ablauf des Erkennungs- und Betätigungsprozesses 12](#_Toc515100015)

[4 Prototypische Umsetzung 12](#_Toc515100016)

[4.1 Hardware 13](#_Toc515100017)

[4.1.1 Verwendete Komponenten 13](#_Toc515100018)

[4.1.2 Schnittstellen 13](#_Toc515100019)

[4.1.3 Aufbau des Prototypen und des Tastermodells 13](#_Toc515100020)

[4.2 Software 13](#_Toc515100021)

[4.2.1 Realisierung der Bildverarbeitungsfunktionen 13](#_Toc515100022)

[4.2.2 Firmware zur Bewegungssteuerung 13](#_Toc515100023)

[4.2.3 Schnittstellen 13](#_Toc515100024)

[4.2.4 Programmstruktur 13](#_Toc515100025)

[4.3 Ergebnisse der Prototypenentwicklung 13](#_Toc515100026)

[5 Fazit und Ausblick 13](#_Toc515100027)

[6 Literaturverzeichnis 15](#_Toc515100028)

# Einleitung

Evtl. ohne unterkapitel

## Ausgangssituation und Problemstellung

## Abgrenzung der Themenstellung

…

Grenzen Stand der Technik

Ziele was wie womit

Was soll genau gemacht werden

Einsatzumgebung

Skizze Aufbau! (oder in 3)

Arbeitsschritte

Mensch-Maschinen-Interaktion hier nicht

Auch nicht Mobilität (autonome Stromversorgung bzw. so konzipiert, dass Stromverbrauch gering und später autonom möglich)

## Aufbau der Arbeit

# Grundlagen

In diesem Kapitel erfolgt eine kurze Einführung in die theoretischen Grundlagen der vorliegenden Arbeit, zudem wird die Thematik auf relevante Bereiche eingegrenzt. Der aktuelle Stand der Technik dieser wissenschaftlichen Themengebiete dient als Ausgangspunkt für die Konzeption im folgenden Kapitel.

Da sich die Arbeit mit der Konzeption einer teilautonomen Funktion für einen Assistenzroboter befasst, soll zunächst die technologische Ausgangsituation in diesem Bereich ermittelt werden. Anschließend erfolgt eine Einführung in die für die Umsetzung der Arbeit wichtigen Themengebiete Mustererkennung sowie Visual Servoing.

## Robotik im Bereich teilautonomer technischer Assistenzsysteme

Technische Assistenzsysteme sind allgemein Systeme, die dem Menschen bei der Bewältigung einer Aufgabe helfen bzw. eine bestimmte Tätigkeit erst ermöglichen. Eine einheitliche Definition des Begriffes fehlt bisher, stattdessen erfolgen Abgrenzungen je nach Themengebiet und Ziel. So definiert das Bundesministerium für Gesundheit in einer Studie technische Assistenzsysteme als Hilfsmittel zur Unterstützung pflegebedürftiger Menschen auf der Basis von Informations- und Kommunikationstechnologien, wozu auch mechatronische Systeme gezählt werden [1]. In dieser Arbeit sollen Systeme aus der Robotik betrachtet werden, welche hilfsbedürftige Menschen teilautonom bei der Bewältigung von Alltagsaufgaben unterstützen.

Roboter lassen sich nach Einsatzgebiet bspw. in Industrie-, Service-, Medizin‑, Erkundungs- oder Militärroboter unterteilen [2]. Serviceroboter werden zur Unterstützung des Menschen derzeit hauptsächlich im Haushalt eingesetzt, dringen aber zunehmend auch in den Bereich der Pflege vor. Nach DIN EN ISO 8373 wird auch bei gleicher Bauform und Funktionsweise zwischen Service- und Industrierobotern nach Einsatzgebiet unterschieden [3]. Im Folgenden sollen Roboter in Anlehnung an die VDI Richtlinie 2860 als frei programmierbare Bewegungsautomaten verstanden werden, die mit Hilfe eines Effektors Handhabungsaufgaben erfüllen können [4]. Bei teilautonomen Systemen sind i.d.R. Sensoren zur selbsttätigen Programmadaption vorhanden.

Ein Beispiel für einen in der Rehabilitation verwendeten Roboter ist der JACO Roboterarm der Firma Kinova Robotics. Dieser ist für eine Montage am Rollstuhl vorgesehen, um Menschen mit eingeschränkter Mobilität mehr Selbständigkeit im Alltag zu ermöglichen [5]. Gesteuert wird der JACO manuell per Joystick, Touchpad bzw. Kopf- oder Kinnsteuerung. Einfache Bewegungsabläufe können auch über eine USB-Schnittstelle programmiert werden. Es fehlt eine Sensorik für eine Adaption zur Laufzeit, daher sind teilautonome Funktionen mit dem JACO nicht zu realisieren.

Im aktuellen Forschungsprojekt SMART-Assist entwickelt das Institut für Robotik und Mechatronik des Deutschen Zentrums für Luft- und Raumfahrt (DLR) das Assistenzrobotiksystem EDAN (EMG-controlled daily assistant) [6]. Es besteht aus einem handelsüblichen elektrischen Rollstuhl, auf welchen ein DLR-Leichtbauroboterarm montiert ist. Schwerpunkte des Projekts sind eine auf Elektromyographie (EMG) basierende Steuerung, sichere Mensch-Maschine-Interaktion sowie adaptives Greifen. Der Sicherheitsaspekt wird durch den Einsatz von flexiblen, adaptiven Materialien (Soft Robotics Konzept) sowie eine Drehmomentregelung erreicht. Integrierte Teilautonomie soll Unterstützung bei Aufgaben des Alltagslebens bieten[[1]](#footnote-1). Veröffentlichungen zur Vorgehensweise oder Ergebnissen bei der Entwicklung von teilautonomen Funktionen existieren bislang nicht.

Ein ähnliches System ist der an der Universität Bremen entwickelte Assistenzroboter FRIEND (Functional robot arm with user-friendly interface for disabled people), welcher als Basis verschiedener Forschungsprojekte dient. Der Aufbau des Systems ist in Abbildung 1 zu sehen.



Abbildung : Assistenzsystem FRIEND der Universität Bremen [7]

Ein Schwerpunkt des ersten Forschungsprojekts ist das Ermöglichen eines teilautomatischen Greifvorgangs. Anstatt wie bspw. beim JACO jede Bewegung vorzugeben, wird hier der komplette Greifprozess mit zwei bis drei Nutzerkommandos initiiert und durchgeführt [7]. Dazu wird zunächst die Kamera am Effektor vom Nutzer in die Nähe des gewünschten Objekts manövriert, anschließend erfolgt mit Hilfe von Visual Servoing (vgl. Abschnitt 2.3) die teilautomatische Ansteuerung des Ziels. Als Einschränkung müssen die zu greifenden Objekte auf einer zum System gehörenden Ablage platziert werden (vgl. Smart Tray in Abbildung 1), welches über Sensoren Gewicht und Position des Objekts bestimmt. Im aktuellen Modellprojekt ReIntegraRob wird durch Verbesserung der Bildverarbeitung ein Arbeiten ohne Smart Tray ermöglicht. Ziel ist Reintegration einer tetraplegisch gelähmten Frau in ihren Beruf als Bibliothekarin [8]. Hierzu soll der Roboter Funktionen wie das Aufnehmen und Ablegen des Buches oder das Umblättern von Seiten autonom durchführen.

Es existieren zahlreiche weitere Industrie- und Forschungsprojekte, welche sich mit Reha- bzw. Assistenzrobotern für Menschen mit Behinderung befassen[[2]](#footnote-2). Bezüglich teilautonomer Funktionalität scheinen die hier vorgestellten Systeme EDAN und FRIEND am aktuellsten bzw. weitesten fortgeschritten. Forschungsbedarf ist allerdings weiterhin vorhanden, da bisher Unterstützung nur für wenige spezielle Aufgaben bzw. Tätigkeiten angeboten wird. Weiterhin ist die Verfügbarkeit der genannten Systeme problematisch, da sie entweder für den normalen Nutzer kaum erschwinglich[[3]](#footnote-3) sind oder ausschließlich in Forschungsprojekten zur Verfügung stehen.

## Mustererkennung als Teilgebiet des maschinellen Sehens

„Computer Vision, auch maschinelles Sehen genannt, umfasst verschiedene Methoden zur Erfassung, Verarbeitung, Analyse und Interpretation von Bildern.“ Mustererkennung in der Bildverarbeitung ist der Prozess zur Identifikation von Objekten. Teilschritte sind Bildaufnahme, Vorverarbeitung, Segmentierung, Merkmalsextraktion, [Klassifizierung](https://de.wikipedia.org/wiki/Klassifizierung) und Aussage bzw. Interpretation [9]. Heutige Standardmethoden zur Mustererkennung verfolgen i.d.R. einen statistischen Ansatz, bei dem Wahrscheinlichkeiten für die Zugehörigkeit eines Objektes zu verschiedenen Kategorien ermittelt werden. [10]

Die Bildvorverarbeitung hat zur Aufgabe, unerwünschte Signalbestandteile zu reduzieren (z. B. Rauschreduktion). Mit Hilfe von Segmentierung werden zusammenhängende bzw. zusammengehörende Regionen in Bildern identifiziert. Anhand von Merkmalen[[4]](#footnote-4) wie bspw. Kanten oder Eckpunkten wird in der Bildanalyse versucht, auf Bildinhalte zu schließen. Das Auffinden verschiedener Merkmale übernimmt die Merkmalsextraktion. Diese liefert für ein Bild einen Merkmalsvektor im Merkmalsraum. Die Klassifikation soll abschließend anhand des Merkmalsvektors die segmentierten Regionen oder Objekte in verschiedene Klassen einteilen [11].

In der visuellen Objekterkennung muss zwischen der Bildklassifikation und der Objektlokalisierung unterschieden werden. Die Bildklassifikation prüft, ob bzw. mit welcher Wahrscheinlichkeit in einem Bild mindestens ein Beispiel einer oder mehrerer Klassen existiert. Das ganze Bild wird der Klasse bzw. Kategorie mit der höchsten Wahrscheinlichkeit zugeordnet. Ziel der Objektlokalisierung (auch Objektdetektion genannt) ist das Auffinden von Objekten verschiedener Klassen inklusive der Bestimmung von Position und Ausdehnung. Für beide Aufgabenstellungen werden heutzutage häufig Verfahren des maschinellen Lernens (Machine Learning) oder des Deep Learnings eingesetzt. Während beim Machine Learning ein vorgegebener Algorithmus zur Extraktion fest vorgegebener Merkmale verwendet wird, erlernt das System beim Deep Learning nach und nach selbst, relevante Merkmale zu extrahieren [11].

Ein bekanntes Beispiel für ein Machine Learning Verfahren zur Objektdetektion ist die Methode nach Viola und Jones [12]. Als mögliche Merkmale für einen gegebenen Bildausschnitt werden hier die Differenzen von Grauwertsummen innerhalb verschiedener Rechtecktypen verwendet, sogenannte Haar-ähnliche Merkmale (vgl. Abbildung 2).



Abbildung : Vier Grundtypen zur Merkmalsberechnung nach Viola und Jones in verschiedener Skalierung und Position [12]

Da die Rechtecktypen in verschiedener Größe und Skalierung verwendet werden, ergibt sich eine hohe Anzahl an möglichen Merkmalen für die Klassifikation. Die Selektion der relevanten Merkmale wird anhand eines sogenannten Boosting-Verfahrens anhand von positiven und negativen Trainingsdaten erlernt. Ein weiterer Schritt zur Effizienzsteigerung ist die Verwendung einer Kaskade von Klassifikatoren, welche negative Bildausschnitte schnell verwirft [12]. Auf der Viola-Jones-Methode basieren zahlreiche weitere Verfahren, bspw. erreicht das Verfahren von Ahonen et al. durch die Verwendung von Local Binary Patterns als texturbasierte Merkmale eine schnellere Trainings- und Detektionszeit [13].

In einer für die Themenstellung interessanten Arbeit werden zunächst verschiedene Merkmale in einem ähnlichen Verfahren wie der Viola-Jones-Methode zur Detektion von unbekannten Fahrstuhltastern verwendet. In anschließenden Nachbearbeitungsschritten werden mögliche Falschdetektionen und nicht erkannte Taster weitgehend korrigiert. Diese Schritte basieren auf der Einschränkung auf solche Taster, die den ADA Richtlinien[[5]](#footnote-5) [14] entsprechen. Unter anderem muss sich dafür die Beschriftung immer (zumindest zusätzlich) links außerhalb des Tasters befinden. Auf der Bedientafel eines Fahrstuhls ergibt sich somit eine Art Gittermuster, welches für die Erkennung genutzt wird (vgl. Abbildung 3). Weiterhin wird im Systemaufbau der Arbeit ein Laser Scanner verwendet, mit welchem die zu erwartende Tastergröße (bekannt aus den ADA Richtlinien) geschätzt werden kann. Bei einem Testdatensatz mit 150 Bildern von Bedientafeln wird eine Erkennungsrate von 86,2 % erreicht, Angaben zur Ausführungsdauer fehlen allerdings [15].

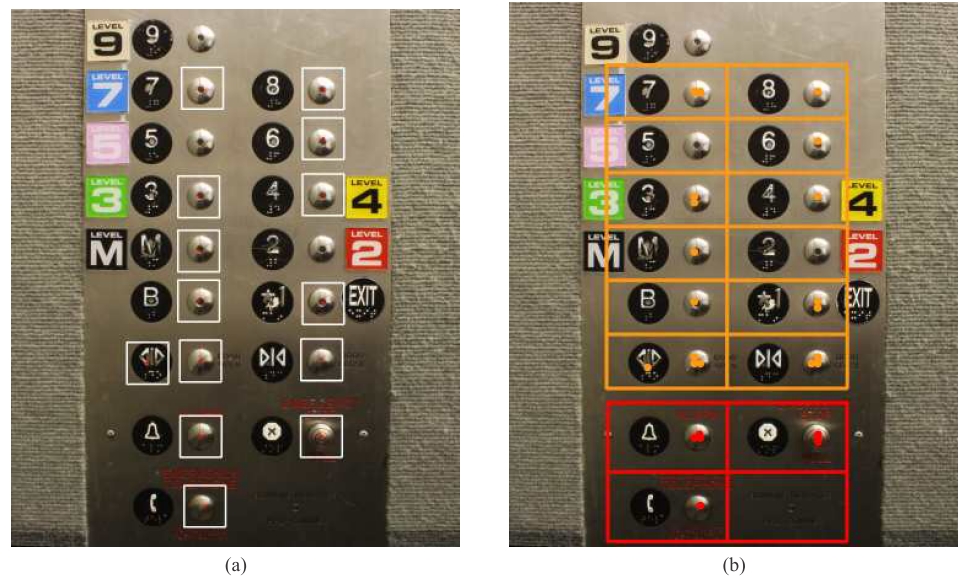


Abbildung : Erkannte Taster (a) und zugehöriges ermitteltes Gitter (b) [15].

Seit dem ersten Sieg von Krizhesvky et al. beim ImageNet Wettbewerb sind Convolutional Neural Networks (CNNs) bei der Bildklassifikation ungeschlagen [16]. Bei CNNs handelt es sich um „faltende“ neuronale Netze, bei denen in jeder Faltungsschicht die Werte eines Filterkernels (Faltungsmatrix) selbständig durch Fehlerrückführung erlernt werden. Eine detaillierte Beschreibung kann bspw. in [17] gefunden werden. Auch der komplexere Bereich der Objektdetektion wird mittlerweile von verbesserten CNN-Architekturen dominiert. Ein erster Durchbruch gelang Girshick et al. mit sogenannten Region-based Convolutional Neural Networks (R-CNNs) [18]. Während vorherige Ansätze meist nach dem Sliding-Windows Prinzip[[6]](#footnote-6) vorgehen, erfolgt bei R-CNNs die Anwendung der neuronalen Netze zur Klassifikation nur auf einer wesentlich geringeren Anzahl von vorgefilterten Regionen. Dadurch kann die Trainings- und Detektionszeit erheblich verkürzt werden[[7]](#footnote-7). [19]

Nach dem ersten Erfolg des R-CNN-Ansatzes zur Objektlokalisierung versuchen Wissenschaftler die Laufzeiten oder die Performanz im Sinne von Erkennungs- und Fehlerraten zu verbessern. Die bislang erfolgreichsten Ansätze sind Faster R-CNNs, You Only Look Once (YOLO) und Single Shot Detector (SSD) mit MobileNet-Architektur[[8]](#footnote-8). Während YOLO bislang die schnellsten Ausführungszeiten vorweisen kann, erzielt SSD bei ebenfalls geringen Laufzeiten höhere Erkennungsraten [20].

Zur Laufzeit von Objekterkennungsverfahren in eingebetteten Systemen bzw. auf Low-Power-Hardware existieren nur wenige Veröffentlichungen. Eine aktuelle Arbeit vergleicht Laufzeiten und Erkennungsraten einiger der oben genannten sowie eines selbst entwickelten Deep Learning Verfahrens bei Ausführung auf einem Raspberry Pi 3 Einplatinencomputer. Bei allen getesteten Verfahren liegt die Bearbeitungszeit bei unter einem Bild pro Sekunde[[9]](#footnote-9). Höhere Erkennungsraten korrelieren mit schlechteren Laufzeiten [21]. Für echtzeitfähige Systeme zur Objektdetektion sind je nach Einsatzgebiet und Anforderungen noch weitere Verbesserungen notwendig.

## Visual Servoing

Aufgabe von Visual Servoing ist es, mit Hilfe von Bildverarbeitungsdaten die Bewegungen oder die gewünschte Position des Effektors eines Roboters zu steuern. Diese Daten können von einer am Manipulator befestigten Kamera stammen (eye-in-hand), wobei Bewegungen des Roboters auch Bewegungen der Kamera auslösen. Eine weitere Möglichkeit ist die Platzierung der Kamera außerhalb des Roboters, sodass dessen Bewegungen von einem festen Punkt aus beobachtet werden (eye-to-hand). Unterschieden wird weiterhin zwischen bildbasierter Regelung (Image Based Visual Servoing), bei welcher Bildinformationen zur direkten Steuerung der Gelenke des Roboters verwendet werden, und positionsbasierter Regelung (Position Based Visual Servoing), wo zunächst eine geometrische Interpretation der Bilddaten erfolgt und daraus ein Steuerbefehl im kartesischen Raum erzeugt wird [22]. Bei der bildbasierten Regelung wird die Regelabweichung aus der Lageabweichung eines vorher aufgenommenen Referenzbilds und der aktuellen Kameraaufnahme ermittelt. Für die positionsbasierte Regelung wird die Stellung der kinematischen Kette im Raum anhand von Bilddaten geschätzt und ein Positionsfehler als Regelabweichung verwendet [23].

Ein stabiler geschlossener Regelkreis zur Steuerung der Gelenke benötigt eine hohe Regelfrequenz. Daten aus Bildverarbeitungssystemen stehen allerdings häufig mit geringerer Frequenz (Bildrate) zur Verfügung, so dass die meisten Visual Servoing Systeme eine positionsbasierte Regelung anwenden [23].

Sowohl zur Positionsschätzung (positionsbasierte Regelung) als auch zur Bestimmung der Lageabweichung zum Referenzbild (bildbasierte Regelung) werden Merkmale (vgl. Abschnitt 2.2) aus der Bildverarbeitung verwendet. Die Transformation zwischen zwei Bildern kann anhand ausgewählter Merkmalspunkte berechnet werden. Werden die Merkmalspunkte markerbasiert ermittelt, führt dies zu einer schnellen und robusten Merkmalsextraktion. Dagegen ist die Verwendung von Merkmalen, die aus dem Zielobjekt generiert werden, zwar aufwendiger aber auch wesentlich flexibler einsetzbar. Zur Bestimmung der Merkmalspunkte liefert zunächst ein Merkmalsdetektor markante Merkmale im Bild, anschließend versieht sie ein Merkmalsdeskriptor mit zusätzlichen Informationen, wie bspw. Orientierung und Umgebung des Punkts. Brauchbare Merkmalspunkte sollten u. a. lokal, möglichst invariant gegenüber Skalierung, Translation und Rotation, wenig anfällig für Rauschen und Verwischen, gut unterscheidbar und effizient zu berechnen sein [24].

Ein bekanntes Verfahren zur Detektion und Beschreibung lokaler Merkmalen ist SIFT (Scale-Invariant Feature Transform), welches die oben genannten Kriterien weitgehend erfüllt. Eine effiziente Abwandlung von SIFT ist das SURF-Verfahren (Speeded Up Robust Features), welches ähnlich robuste und invariante Merkmale in kürzerer Berechnungszeit liefert [25].

Ein Vergleich zwischen verschiedenen Bildern desselben Objekts liefert zusammengehörige Merkmalspunkte (auch Feature Matching genannt). Sind die Bilder aus unterschiedlichen Perspektiven entstanden, kann anhand dieser Merkmalspunkte die Homographie[[10]](#footnote-10) und damit die Bewegung der Kamera (oder des Objektes)berechnet werden [24].

Wo angewendet, Grenzen / Lücken

Paper: Vergleich verschiedener bildbasierter Regler zur Realisierung teilautonomer Greifvorgänge ?

# Konzeption

Zunächst klären, was vorhanden, womit gearbeitet werden kann -> Eingrenzung der Möglichkeiten, teilweise Restriktionen für Teilaufgaben.

Unsicherheiten, mögliche Fehler früh erwähnen

Dann Lösungen zu Teilaufgaben erarbeiten

**Skizze Aufbau! (oder in 1)**

## Manipulator mit Bildverarbeitungseinheit

Wozu

Ziel, Aufgabe, Restriktionen

Siehe Aktorsteuerung und Objektverfolgung (PDF)

### Anforderungen an das System

Nicht zu komplex -> 3 DOF

Anforderungen (kamera?)

### Kriterien für die Auswahl der Komponenten

Preis, Verfügbarkeit

Einfach zu implementieren

Leistung muss ausreichen

### Aufbau des Systems

Komponenten, Controller, Treiber (hier?) etc.

Was gibt es (Ärme)? Schwächen: Keine oder rudimentäre Software etc.

Mehr „Spielzeug“

Kamera, Effektor (mit Sensor), Manipulator (mit Motoren), Controller (mit Treibern), Computer für Programmlogik (evtl. getrennt von Controller Vor-/ Nachteile) getrennte Funktionsbereiche (Steuerung Aktor, Programmlogik)

## Objekt- und Texterkennung

Zwei Verfahren -> Test Geschwindigkeit auf vorhandener Hardware

Erst Taster detektieren, dann Text auslesen

### Maschinelles Detektionsverfahren mit vorgegebenem Merkmalsraum

### Deep Learning Ansatz zur Detektion ohne Merkmalsraumvorgabe

… (Convolutional Neural Networks)

### Auslesen der Tasterbeschriftung

Textdetektion (nicht recognition) als Herausforderung

Recognition im Prinzip Klassifikation (jedes Zeichen ein Bild)

MNIST, char74k etc.

Alle Ansätze

## Objektverfolgung

Ziel / Unterschied bewegter Hintergrund

Hier Teil des Visual Servoing

Nur Translation interessant?

### Echtzeit-Objektverfolgung in Videosequenzen

Wäre einfacher (?) bzw. vorzuziehen (warum?)

### Merkmalsabgleich und perspektivische Transformation zwischen Einzelbildern

## Entfernungsberechnung

2D Bild -> 3D Informationen

### Berechnung durch Änderung des Winkels zum Ziel

### Berechnung durch Änderung der Bildgröße des Ziels

## Bewegungssteuerung

Hier Teil des Visual Servoing

### Wahl des Koordinatensystems

### Positionierung des Effektors im Raum

Inverse Kinematik…

## Integration der Teillösungen

### Benutzerschnittstelle

### Interaktion der Komponenten

Kommunikation

Befehlssatz (oder in 4) (dann im Anhang)

### Ablauf des Erkennungs- und Betätigungsprozesses

# Prototypische Umsetzung

Implementierung und Auswertung der Teillösungen

Modell!

Verfahren zur Tasterdetektion

Verwendete Bibliothek(en)

Wie womit implementiert

Auswertung / Gegenüberstellung der Verfahren

Zielverfolgung bei Kamerabewegungen

Firmware zur Bewegungssteuerung des Manipulators

Ansteuerung der Motoren

Kommunikation und Befehlssatz

Erkennung einer Tasterbetätigung

Endlagenschalter

Optional „nachgiebiger Finger“, der erst ab bestimmter Krafteinwirkung auslöst

Implementierung

Firmware Manipulator

Tests etc.

Test des Gesamtsystems

## Hardware

### Verwendete Komponenten

Endlagenschalter

### Schnittstellen

### Aufbau des Prototypen und des Tastermodells

## Software

### Realisierung der Bildverarbeitungsfunktionen

### Firmware zur Bewegungssteuerung

Mit Befehlssatz (oder in nächsten Abschnitt)

Erkennung Tasterdetektion

### Schnittstellen

Benutzer und Systemintern (nach außen und innen)

### Programmstruktur

## Ergebnisse der Prototypenentwicklung

# Fazit und Ausblick

# Literaturverzeichnis

|  |  |
| --- | --- |
| [1] | Bundesministerium für Gesundheit, „Abschlussbericht zur Studie: Unterstützung Pflegebedürftiger durch technische Assistenzsysteme,“ VDI/VDE Innovation + Technik GmbH, IEGUS – Institut für Europäische Gesundheits- und Sozialwirtschaft GmbH, Berlin, 2013. |
| [2] | T. Christaller, M. Decker, J. Gilsbach und G. Hirzinger, Robotik - Perspektiven für menschliches Handeln in der zukünftigen Gesellschaft, Berlin Heidelberg: Springer-Verlag , 2001. |
| [3] | DIN Deutsches Institut für Normung e.V., *Roboter und Robotikgeräte - Wörterbuch (ISO/DIS 8373:2010); Entwurf November 2010,* Berlin: Beuth Verlag GmbH, 2010. |
| [4] | VDI, „VDI Richtlinie 2860: Montage- und Handhabungstechnik; Handhabungsfunktionen, Handhabungseinrichtungen; Begriffe, Definitionen, Symbole,“ VDI, Düsseldorf, 1990. |
| [5] | Maheu et al., „Evaluation of the JACO robotic arm: Clinico-economic study for powered wheelchair users with upper-extremity disabilities,“ in *IEEE international conference on rehabilitation robotics (ICORR)*, Zürich, 2011. |
| [6] | Vogel et al., „An assistive decision-and-control architecture for force-sensitive hand–arm systems driven by human–machine interfaces,“ *The International Journal of Robotics Research (IJRR) , vol. 34, no. 6,* pp. 763-780, Mai 2015. |
| [7] | C. Martens, O. Prenzel und A. Graese, „The Rehabilitation Robots FRIEND-I & II: Daily Life Independency through Semi-Autonomous Task-Execution,“ in *Rehabilitation Robotics, Sashi S Kommu*, IntechOpen, 2007. |
| [8] | Universität Bremen - Institute of Automation, „Assistenzroboter FRIEND,“ 16 Januar 2014. [Online]. Available: http://www.iat.uni-bremen.de/sixcms/detail.php?id=555. [Zugriff am 18 Mai 2018]. |
| [9] | L. Priese, Computer Vision - Einführung in die Verarbeitung und Analyse digitaler Bilder, Berlin Heidelberg: Springer Vieweg, 2015. |
| [10] | M. Hassaballah, A. Abdelmgeid und A. Hammam, „Image Features Detection, Description and Matching,“ in *Image Feature Detectors and Descriptors: Foundations and Applications - Studies in Computational Intelligence (Book 630)*, Switzerland, Springer International Publishing, 2016, pp. 11-45. |
| [11] | H. Süße und E. Rodner, Bildverarbeitung und Objekterkennung - Computer Vision in Industrie und Medizin, Wiesbaden: Springer Vieweg, 2014. |
| [12] | P. Viola und M. Jones, „Robust Real-time Object Detection,“ in *SECOND INTERNATIONAL WORKSHOP ON STATISTICAL AND COMPUTATIONAL THEORIES OF VISION – MODELING, LEARNING, COMPUTING, AND SAMPLING*, Vancouver, 2001. |
| [13] | A. T., H. A. und P. M., „Face Recognition with Local Binary Patterns,“ in *Computer Vision - ECCV 2004. ECCV 2004. Lecture Notes in Computer Science, vol 3021*, Berlin, Heidelberg, Springer-Verlag, 2004. |
| [14] | American with Disability Act, „ADA Compliance Directory - ELEVATORS,“ 2018. [Online]. Available: http://www.ada-compliance.com/ada-compliance/ada-elevators.html. [Zugriff am 12 05 2018]. |
| [15] | E. Klingbeil, B. Carpenter, O. Russakovsky und A. Y. Ng, „Autonomous operation of novel elevators for robot navigation,“ in *2010 IEEE International Conference on Robotics and Automation*, 2010. |
| [16] | A. Krizhevsky, I. Sutskever und G. E. Hinton, „ImageNet Classification with Deep Convolutional Neural Networks,“ in *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou und K. Q. Weinberger, Hrsg., Curran Associates, Inc., 2012, pp. 1097-1105. |
| [17] | Y. Lecun, L. Bottou, Y. Bengio und P. Haffner, „Gradient-based learning applied to document recognition,“ in *Proceedings of the IEEE*, 1998. |
| [18] | R. B. Girshick, J. Donahue, T. Darrell und J. Malik, „Rich feature hierarchies for accurate object detection and semantic segmentation,“ *CoRR,* Bd. abs/1311.2524, 2013. |
| [19] | A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto und H. Adam, „MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications,“ *CoRR,* Bd. abs/1704.04861, 2017. |
| [20] | J. Hui, „Object detection: speed and accuracy comparison (Faster R-CNN, R-FCN, SSD, FPN, RetinaNet and YOLOv3),“ 27 März 2018. [Online]. Available: https://medium.com/@jonathan\_hui/object-detection-speed-and-accuracy-comparison-faster-r-cnn-r-fcn-ssd-and-yolo-5425656ae359. [Zugriff am 20 04 2018]. |
| [21] | Y. Zhang, H. Peng und P. Hu, „CS341 Final Report: Towards Real-time Detection and Camera Triggering,“ in *Project in Mining Massive Data Sets*, 2017. |
| [22] | F. Chaumette und S. Hutchinson, „Visual servo control, Part I: Basic approaches,“ *IEEE Robotics and Automation Magazine,* Bd. 13, pp. 82-90, 2006. |
| [23] | Assistive Innovations, „iARM,“ Assistive Innovations, [Online]. Available: https://www.assistive-innovations.com/de/roboterarme/iarm-de. [Zugriff am 18 Mai 2018]. |

1. Als Beispiel in Schaubildern oder Videos dient die Hilfe beim Trinken. [↑](#footnote-ref-1)
2. Ein Beispiel für ein verfügbares System ist der iARM der Firma Assistive Innovations [19], welcher ähnliche Funktionen wie der JACO bietet. [↑](#footnote-ref-2)
3. Die Preie von JACO und iARM liegen im Bereich von mehreren Zehntausend Euro. [↑](#footnote-ref-3)
4. Ein Merkmal ist eine numerische oder qualitative Information, die aus Bildpunkten einer Region (lokal) oder des gesamten Bildes (global) berechnet wird. Merkmale können bspw. farb‑, form- oder texturbasiert sein, aber auch durch abstrakte Operationen entstehen [10]. [↑](#footnote-ref-4)
5. In Deutschland entspricht nur ein Teil der Fahrstuhltaster diesen Richtlinien. [↑](#footnote-ref-5)
6. Beim Sliding-Window-Prinzip wird ein Fenster vordefinierter Größe Pixel für Pixel über das gesamte Bild verschoben und an jeder Position eine Klassifikation vorgenommen [11]. [↑](#footnote-ref-6)
7. Resultat ist eine Reduktion der Detektionszeit auf ca. 50 Sekunden auf einer High-End-CPU des Jahres 2013 [16]. [↑](#footnote-ref-7)
8. Insbesondere hat auch die Architektur der verwendeten tiefen neuronalen Netze, wie z. B. Anzahl und Typ der Schichten, einen Einfluss auf Erkennungsrate und Ausführungszeiten. MobileNets eignen sich für mobile Geräte und erzielen trotz kompakter Größe dennoch hohe Erkennungsraten [17]. [↑](#footnote-ref-8)
9. Die Detektionsrate liegt im Bereich von 0,1 fps bis 0,7 fps (frames per second) [↑](#footnote-ref-9)
10. Zu geometrischen Transformationen und homogenen Koordinaten siehe bspw. [11]. [↑](#footnote-ref-10)