

---

# Augmented Reality basierte Roboterzustands- und Prozessvisualisierung am Beispiel aus der industriellen Automatisierung

---

Augmented Reality-based Robot State and Process Visualisation on an Industrial Process  
Automation Example  
Master-Thesis von Anna Grabiec



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT

Fachbereich Informatik  
Fachgebiet Simulation, Systemopti-  
mierung und Robotik

Augmented Reality basierte Roboterzustands- und Prozessvisualisierung am Beispiel aus der industriellen Automatisierung  
Augmented Reality-based Robot State and Process Visualisation on an Industrial Process Automation Example

Vorgelegte Master-Thesis von Anna Grabiec

Prüfer: Prof. Dr. Oskar von Stryk  
Betreuer: M.Sc. Jérôme Kirchhoff

Tag der Einreichung: 18. August 2016

---

---

Erklärung zur Master-Thesis

---

Hiermit versichere ich, die vorliegende Master-Thesis ohne Hilfe Dritter nur mit den angegebenen Quellen und Hilfsmitteln angefertigt zu haben. Alle Stellen, die aus Quellen entnommen wurden, sind als solche kenntlich gemacht. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Darmstadt, den 18. August 2016

---

(Anna Grabiec)



---

## Zusammenfassung

---

In vielen neuartigen Ansätzen zur industriellen Automatisierung wird eine enge Zusammenarbeit von Mensch und Roboter vorausgesetzt. Durch die Nutzung des gemeinsamen Arbeitsraums, können Produktionsabläufe effektiver gestaltet werden. Dies erfordert eine missverständnisfreie Kommunikation zwischen dem Roboter und dem Arbeiter. Die meisten Industrieroboter können dabei nur begrenzt mit der Umgebung kommunizieren. Statt dem Roboter das Verständnis für die aktuelle Situation beizubringen, ist der Ansatz, in welchem der Mensch die Intention und den Betriebszustand des Roboters verstehen lernt, intuitiver.

In dieser Arbeit wurde eine Kommunikationslösung zwischen dem Roboter und Mensch vorgestellt, welche dem Menschen als Hilfsmittel zur Interaktion mit dem Industrieroboter zur Seite gestellt werden kann.

Es wurde ein Konzept der graphischen Darstellung von der Roboterinformation und -prozess unter Verwendung von Augmented Reality (AR) beschrieben und realisiert. Dazu wurde ein Verfahren zur Kamera-Roboter Kalibrierung entwickelt, welches die Positionierung beliebig vieler Marker auf dem gewählten Roboterglied gestattet. Dabei ist die gleichzeitige Änderung der Lage des Aufnahmegerätes und die Bewegung des Roboterarmes (was bei den üblichen AR System nicht gegeben ist) erlaubt.

Da kein Fachwissen zur Nutzung der entwickelten Applikation vorausgesetzt sollte, wurde eine intuitive und AR basierte Benutzerschnittstelle entwickelt.

Das vorgestellte System wurde im Bezug auf die Darstellung an einem konkreten Beispiel eines Leichtbauroboters, welcher für sichere Mensch-Roboter Kollaboration in der industriellen Automatisierung entwickelt wurde, bewertet.

Des Weiteren wurde der Einfluss der in der Benutzeroberfläche dargestellten Informationen auf den Anwender über eine Benutzerstudie analysiert. Es hat sich gezeigt, dass das entwickelte Konzept zur Visualisierung des Roboterzustandes und -prozesses eine positive Wirkung auf die Roboter-Mensch Kommunikation hat und zur Steigerung des Situationsverständnisses des Anwenders beiträgt.



---

## Abstract

---

Since industrial equipment is highly automatized, a close cooperation between human and a machine is required. Because of the application of robots in a mutual workspace, it became possible to advance the production processes. An optimization of reliable communication between robots and workers is essential for further improvement. Most of the industrial robots are able to achieve only sparse contact with the environment. It is plausible to teach the workers how the robots proceed. If the intentions and course of actions of those machines can be intuitively understood by workers, it can lead to simplifications of the operational procedures. In this study an all-in system was designed, which provides a tool for the workers in the interaction with industrial robots. It is addressed at both experts and new users. Furthermore the user can, by the use of mobile devices, like tablet computer or smartphone, asses simultaneously the actions of the robot. The progression of the movement can also be observed and determined in a new perspective. A draft of graphical description for robot intentions and processes by the use of Augmented Reality (AR) was established and realized.

Furthermore a method for camera robot calibration was developed, allowing determining position changes of the robot arm and concomitant position changes of the recording device, as long as at least one marker is in the field of view.

For the use of this interface no expert knowledge or overview of available robot data are necessary, an intuitive and easy accessible user interface was developed. The presented framework was based on the example of a light weight robot. This robot was developed for secure human – robot collaboration and partly evaluated in this study. The question how the information in the user interface can influence the user was analysed . The obtained results showed that the concept of visualisation of the robot state improved the communication between robot and worker. Furthermore by the use of that approach the understanding of the situation for the user was increased.



---

## Inhaltsverzeichnis

---

<b>1</b>	<b>Einleitung</b>	<b>3</b>
1.1	Motivation . . . . .	3
1.2	Zielsetzung . . . . .	5
1.3	Gliederung der Arbeit . . . . .	6
<b>2</b>	<b>Hintergrund</b>	<b>7</b>
2.1	Augmented Reality - Grundprinzip und Überblick . . . . .	7
2.2	Situationbewusstsein . . . . .	10
2.3	Ergonomische Anwendungsgestaltung . . . . .	13
2.4	Roboter-Mensch Interaktion - Stand der Forschung . . . . .	14
2.4.1	Kommunikation durch Bewegung und Akustik . . . . .	15
2.4.2	Visuelle Kommunikation . . . . .	17
2.5	Bibliotheken und Softwarepakete . . . . .	20
2.5.1	Robot Operating System . . . . .	20
2.5.2	ARToolKit . . . . .	22
2.5.3	OpenCV . . . . .	22
<b>3</b>	<b>Konzepte und Methoden</b>	<b>23</b>
3.1	Problemstellung und Anforderungsanalyse . . . . .	23
3.2	Grundlagen . . . . .	24
3.2.1	Zerlegung einer Transformationsmatrix . . . . .	24
3.2.2	Frame - Relative Lage des Objekts . . . . .	25
3.2.3	Kameramodell . . . . .	26
3.3	Roboterzustand und -prozess . . . . .	29
3.3.1	Statusinformationen . . . . .	30
3.3.2	Intention . . . . .	30
3.3.3	Wartungsinformationen . . . . .	31
3.4	Kalibrierung der AR-Komponente . . . . .	32
3.4.1	Markerbasiertes Tracking . . . . .	34
3.4.2	Marker - Roboter Kalibrierung . . . . .	35
3.4.3	Bestimmung der Kamera-Roboter Relation . . . . .	36
3.5	Software Architektur . . . . .	38
<b>4</b>	<b>Umsetzung</b>	<b>41</b>
4.1	Systemgestaltung . . . . .	41
4.2	Systemkalibrierung und Objektregistrierung . . . . .	43
4.3	Graphische Benutzerschnittstelle . . . . .	45

---

5	Evaluation	49
5.1	Darstellung des Zustands und Prozesses von dem BioRobArm . . . . .	49
5.1.1	Experiment . . . . .	50
5.1.2	Systemkalibrierung . . . . .	50
5.1.3	Roboterintention und -zustand . . . . .	51
5.2	Einfluss der dargestellten Informationen - Benutzerstudie . . . . .	56
5.2.1	Methoden . . . . .	57
5.2.2	Versuchsaufbau . . . . .	58
5.2.3	Versuchsablauf . . . . .	58
5.2.4	Auswertung . . . . .	59
5.3	Diskussion . . . . .	63
6	Zusammenfassung	65

---

## Abbildungsverzeichnis

---

1.1	Menschen arbeiten mit Robotern "Hand in Hand". . . . .	3
1.2	Mensch-Roboter Interaktion . . . . .	4
2.1	Erweiterung der realen Szene mit einem virtuellem Auto <b>Quelle:</b> [19] . . . .	8
2.2	Beispielmarker genutzt bei optisch-markerbasierten Trackingverfahren . .	9
2.3	Registrierung der Objekte <b>Quelle:</b> [20] . . . . .	9
2.4	See Through mit einer Datenbrille (Mini Augmented Vision) <b>Quelle:</b> [1] . .	10
2.5	Situationsbewusstsein - Zusammensetzung <b>Quelle:</b> Modifiziert nach [12] . . . .	11
2.6	Informationsloch <b>Quelle:</b> Modifiziert nach [28] . . . . .	15
2.7	Bewegungsbasierte Roboter-Mensch Kommunikation <b>Quelle:</b> [31] . . . . .	16
2.8	Arbeitsraumüberwachung mit einem kamera- und projektionsbasiertem Sensorsystem <b>Quelle:</b> [34] . . . . .	17
2.9	Fraunhofer-Institut IFF. Arbeitsraumüberwachung <b>Quelle:</b> [35] . . . . .	17
2.10	Baxter - zweiarmiger Roboter mit einem festmontiertem Display. <b>Quelle:</b> [2]	18
2.11	Baxter kommuniziert mit dem Menschen mit Hilfe von Bildzeichen <b>Quelle:</b> [36]	18
2.12	Optima ML 750 - Mobiler Industrietransportroboter <b>Quelle:</b> [38] . . . . .	19
2.13	<i>Ros</i> -Kommunikationsprinzip <b>Quelle:</b> Modifiziert nach [40] . . . . .	20
2.14	Rosbridge Pipeline für eine Webapplikation <b>Quelle:</b> Modifiziert nach [47] . . .	22
3.1	Framekonzept - relative Lage eines Objekts . . . . .	25
3.2	Lochkamera <b>Quelle:</b> Modifiziert nach [54] . . . . .	27
3.3	Kameramodell <b>Quelle:</b> Modifiziert nach [54] . . . . .	27
3.4	Transformation von Welt- über Kamera- in Pixelbildkoordinaten erfolgt mit Hilfe von einer Projektionsmatrix, die die extrinsischen und intrinsi- schen Eigenschaften der Kamera beschreibt. . . . .	28
3.5	Roboterzustand setzt sich aus verschiedenen Sensorsignalen zusammen. .	29
3.6	Zusammenhang der verschiedenen Schritte bei der Bestimmung der Kame- raposition und der Objektregistrierung . . . . .	33
3.7	Markererkennung <b>Quelle:</b> Modifiziert nach [14] . . . . .	35
3.8	Schritte der Marker-Roboter Kalibrierung . . . . .	36
3.9	Bestimmung der ArMarkerposition relativ zum Tragetframe. . . . .	36
3.10	Bestimmung der Kameraposition bezüglich der Roboterbasis . . . . .	37
3.11	Systementwurf. Server-Client-Architektur . . . . .	38
3.12	Entwurf der graphischen Benutzeroberfläche . . . . .	40
4.1	Systemstruktur und Kommunikationsübersicht . . . . .	41
4.2	System Schnittstellenübersicht . . . . .	42
4.3	Die Anordnung der <i>Tf</i> -Frames, die während der Systemausführung stets automatisch bestimmt wird . . . . .	44

---

4.4	Umsetzung der graphischen Benutzeroberfläche . . . . .	46
5.1	Struktur des BioRob Arms <b>Quelle:</b> [3] . . . . .	49
5.2	Bewegungsbereich des BioRob-Arms. (a) Seitenansicht und (b) Draufsicht <b>Quelle:</b> [3] . . . . .	50
5.3	Der Biorob-Arm und seine Arbeitsumgebung. . . . .	51
5.4	Marker-Roboter Kalibrierung am Beispiel des Biorob-Arms . . . . .	52
5.5	Sicht auf die Benutzeroberfläche . . . . .	53
5.6	Visualisierung des BioRob-Arms. Intention mit dem Plug-In . . . . .	54
5.7	Movementschaltfläche . . . . .	54
5.8	Prozessplan des Biorobs. . . . .	55
5.9	Visualisierung des BioRob Zustands. Wartungsinformationen. . . . .	56
5.10	Versuchsaufbau . . . . .	59
5.11	Verteilung der richtigen Antworten während des Experiments . . . . .	60
5.12	Bewertung des aktuellen Roboterzustands . . . . .	61
5.13	Bewertung der Benutzeroberfläche . . . . .	62

---

## Tabellenverzeichnis

---

2.1	Situationsbewusstseinsmodell für den Arbeiter im Kontrollraum einer Roboterzelle. . . . .	12
2.2	Situationsbewusstseinsmodell für den Arbeiter, der mit dem Roboter zusammen arbeitet und keine Kenntnis über dessen inneren Zustand hat. .	12
5.1	Einschätzung des Roboterzustands unter zu Zuhilfenahme der Applikation.	61



---

## 1 Einleitung

---

### 1.1 Motivation

---

In der industriellen Automatisierung werden bis zum Jahr 2018 voraussichtlich 1,3 Millionen Industrie-Roboter im Einsatz sein [4]. Roboter sind heutzutage zu einem maschinellen Partner für den menschlichen Mitarbeiter geworden, durch eine große Verfügbarkeit der Robotertechnik werden sie häufiger auch in kleinen Produktionen eingesetzt.

Neuartige Produktionsanlagen setzen eine enge Zusammenarbeit von Mensch und Roboter voraus. Der Mensch soll mit der intelligenten Maschine „Hand in Hand“ arbeiten können. Durch die Ausnutzung der jeweiligen Stärken können die Produktionsabläufe effektiver gestaltet werden [5]. Monotone, kraftraubende, oder gefährliche Produktionsschritte sollen mit Roboterassistenz ausgeführt werden. Das bedeutet, der Arbeitsbereich des Roboters wird an den Arbeitsbereich des Arbeiters und des Produkts angepasst. Wie in Abbildung 1.1 zu sehen ist, könnte der Roboter zum Beispiel dem Mitarbeiter Objekte zum richtigen Zeitpunkt und der richtigen Position bereit stellen. Durch solch eine Kollaboration kann die Flexibilität, Effizienz, Wirtschaftlichkeit und Qualität der Produktion erhöht werden.

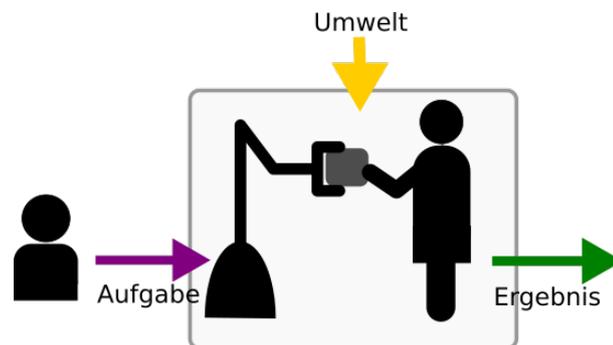


Abbildung 1.1: In modernen Produktionsanlagen arbeiten Menschen mit Robotern "Hand in Hand". Der physische Kontakt ist dabei nicht ausgeschlossen oder sogar gewollt.

Die direkte Interaktion zwischen Mensch und Roboter wird räumlich und zeitlich nicht beschränkt, d.h. die üblichen Schutzvorrichtungen (zum Beispiel Zäune), sind nicht mehr vorhanden [6]. Wenn sich der Mensch und die Maschine einen Arbeitsraum teilen, muss Sorge für die Sicherheit des Menschen getragen werden. Verletzungen an Menschen sind im Gegensatz zur Beschädigung eines Roboters inakzeptabel. Neben Einhaltung der relevanten Normen (ISO TS 1566:2016 [7], ISO 10218-1 [8], ISO 10218-2 [9]) und Richtlinien [10], welche Vorgaben zur maximalen Robotergeschwindigkeit, Abständen zwischen dem Mensch und dem Roboter, oder Arbeitsraumüberwachung behandeln, ist es wichtig,

dass zumindest eine der Parteien (Mensch oder Roboter) die Absichten des anderen versteht und richtig deuten kann. Die Steigerung des Verständnisses für das Vorhaben des Roboters, beziehungsweise des Menschen spielt dabei eine entscheidende Rolle. Dementsprechend ist die Art und Weise auf welche der Roboter und der Mensch kommunizieren ausschlaggebend.

Der Mensch versteht Objekte und seine Mitmenschen durch Interaktion mit ihnen. Er nimmt ein technisches System, mit dem er arbeitet, mittels sensorischer Reize, wie Sehen, Hören oder Tasten wahr [11]. Menschen interpretieren diese Informationen aufgrund ihrer subjektiven Erfahrungen und Wissen [12]. Währenddessen kann der Roboter in den meisten Fällen nur begrenzt mit der Umgebung kommunizieren (z.B. durch visuelle Signale). Für ihn ist es unmöglich seine Umgebung und die aktuelle Situation ohne zusätzliches Wissen zu verstehen. Jedoch kann ein Robotersystem insofern erweitert werden, dass es sein Verhalten an den menschlichen Partner anpasst. Dazu beobachtet und analysiert der Roboter, wie in Abbildung 1.2(a) dargestellt, den Arbeitsraum und den Menschen. Anhand von diesen Informationen kann das technische System entscheiden, welche Aktion (Bewegung) der Mensch durchführen will und darauf reagieren.

Während ein Roboter über eine endliche Anzahl von durchführbaren Aktionen verfügt, ist das menschliche Verhalten weitaus komplexer. Der Roboter muss entscheiden welche Intention die jeweilige Aktion des Menschen mit sich führt. Bei komplexen Situationen kann es daher zu Missverständnissen, oder verzögerten Reaktionen kommen, die potenziell gefährlich für den Menschen sein können. Die durchzuführende, kollaborative Aufgabe kann somit nicht erfolgreich abgeschlossen werden.

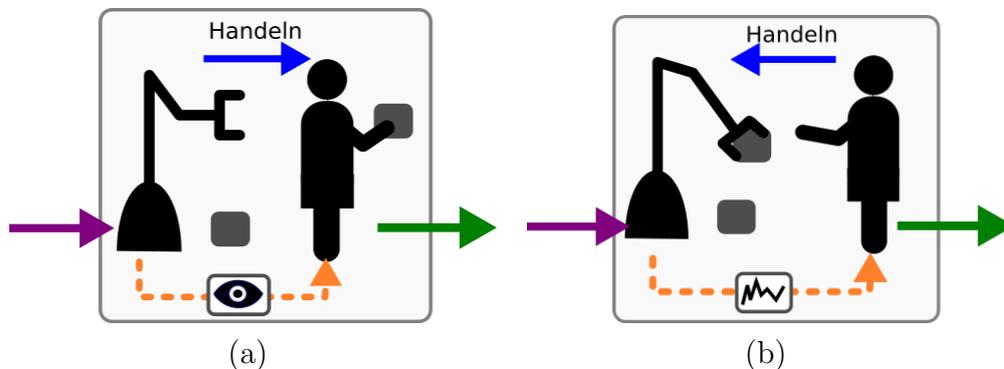


Abbildung 1.2: Mensch-Roboter Interaktion. (a) Der Roboter versteht den Menschen. Er beobachtet und analysiert seine Haltung und passt sein eigenes Verhalten an. (b) Der Mensch versteht den Roboter. Er kann durch das Beobachten und Analysieren des Roboterzustands eigenes Verhalten anpassen und den Umgang mit dem Roboter effizienter und sicherer gestalten.

Viel intuitiver und effizienter ist der Ansatz, in welchem der Mensch die Intention und den Betriebszustand des Roboters zu verstehen lernt, vergleiche Abbildung 1.2(b). Dabei wird die Intention des Roboters durch den aktuellen inneren Zustand des Systems abgebildet. Die Interpretation dieser Daten ist jedoch für ungeübte Benutzer schwierig.

---

Durch eine sinnvolle Informationsbereitstellung kann der Anwender dabei unterstützt werden. Mit der Verwendung von Augmented Reality (AR) Ansätzen können, zum Beispiel Bewegungspfade direkt in der Umgebung des Anwenders dargestellt werden. Die Verwendung von allgemein gebräuchlichen Darstellungsformen, wie Listen und Diagrammen, erleichtert die Zahlinterpretation. Mit der graphischen Darstellung und Auswertung der langfristig aufgezeichneten Messungen kann dem Benutzer ein klares Bild über den Zustand des Roboters übermittelt werden, welches wiederum rechtzeitige Wartungs- und Instandhaltungsroutinen durchzuführen erlaubt und bei Roboterfällen die Fehlersuche unterstützt [13].

Im industriellen Alltag kann die Bereitstellung der Informationen über den Roboterzustand und -prozess eine höhere Effektivität der Produktion, Verstärkung der Sicherheit, sowie des mentalen Komforts der Arbeiter bedeuten.

---

## 1.2 Zielsetzung

---

Ziel dieser Arbeit ist die Erarbeitung eines effizienten und kostengünstigen Kamera-Robotersystems, welches die Intention und den Zustand des Roboters dem Endbenutzer intuitiv vermittelt. Dabei sollen auf Basis der Augmented Reality-Technologie die ausgewählten Roboterdaten benutzerfreundlich visualisiert werden.

Durch eine geeignete Darstellung der Roboterprozesse soll die Roboterleistung überwacht und analysiert werden können. Der erarbeitete Ansatz soll in der betrieblichen Praxis die direkte Mensch-Roboter Kollaboration unterstützen und sicherer gestalten. Zusätzlich soll der Endbenutzer den Roboter besser verstehen lernen.

Ein weiteres Ziel besteht in einer Steigerung der Produktivität (Vorbeugen von Produktionsausfällen) indem Wartungsinformationen geeignet visualisiert werden, was bei einer Verschlechterung der Roboterperformanz während der Ursachenermittlung hilfreich sein kann.

Für die Erarbeitung eines geeigneten Lösungsansatzes sind folgende Schritte erforderlich:

- i Es ist eine geeignete Informationsdarstellung zu finden, um die Roboterintention und -Zustand eindeutig zu übermitteln.
- ii Zur Realisierung der AR-gestützten Darstellung muss ein praxisgerechtes Verfahren zur Ermittlung der Kameraposition bei sich bewegender Kamera und Roboter gefunden werden.
- iii Es ist ein plattformunabhängiges und leistungsfähiges System zu gestalten.
- iv Der entwickelte Ansatz ist an einem realen Robotersystem, sowie über eine Benutzerstudie zu evaluieren.

Die Verwendung der AR, setzt eine exakte Überlagerung der realen Objekte mit den virtuellen Inhalten voraus und somit den Einsatz einer Kamera und eines geeigneten Kalibrierungsverfahrens zwischen Kamera- und Roboter. Die Entwicklung solch eines Kalibrierverfahrens ist ein Teil dieser Arbeit. Dabei sollen keine zusätzlichen Vorrichtungen

---

benötigt werden und bei Änderungen der Roboter- oder Kameraposition keine erneute Kalibrierung erforderlich ist.

Der entwickelte Visualisierungsansatz, sowie die graphische Benutzerschnittstelle des Systems, soll als Grundlage zur Steigerung des Situationsverständnisses dienen und Basis einer richtigen Wahrnehmung des Roboterzustandes und -prozesses sein. Weiterhin wird die Erfüllung dieser Vorgaben über eine Benutzerstudie überprüft, welche neben der Kalibrierung den Mittelpunkt dieser Arbeit bildet.

---

### 1.3 Gliederung der Arbeit

---

Nachdem im Kapitel 1 die Motivation für diese Arbeit und Formulierung der Aufgabenstellung dargestellt wurde, wird im Kapitel 2 das Grundprinzip und der Überblick über die Augmented Reality Technologie beschrieben. Es wird zusätzlich der Begriff des Situationsbewusstseins definiert und dessen Einfluss auf das Verhalten der Menschen verdeutlicht. Als nächstes werden die Anforderungen an die ergonomische Anwendungsgestaltung abgeleitet und die existierenden Ansätze der Roboter-Mensch Interaktion vorgestellt. Abschließend erfolgt ein kurzer Überblick über die Bibliotheken und Softwarepakete, die bei der Implementierung verwendet wurden.

Im Kapitel 3 werden die Problemstellungen und Systemanforderungen erläutert und grundlegenden Definitionen formuliert. Zudem werden Größen, die den Roboterzustand und -prozess beschreiben, festgelegt und Konzepte zu deren Visualisierung vorgestellt. Zusätzlich wird ein Entwurf der Kamera-Roboter Kalibrierung eines AR-Systems beschrieben. Zum Schluss wird ein Überblick über die Software Architektur des Systems gegeben.

Die vorgestellten Konzepte sind in einer Software umgesetzt worden. Die finale Implementierung wird im Kapitel 4 vorgestellt und schließlich im Kapitel 5 diskutiert. Besonderes Augenmerk der Applikationsbewertung liegt hier auf der graphischen Darstellung der Roboterintention und dessen Einfluss auf den Benutzer.

---

## 2 Hintergrund

---

In diesem Kapitel werden die in dieser Arbeit benötigten Grundlagen und die verwandten Ansätze vorgestellt. Als erstes wird das Grundprinzip der Augmented Reality (AR) dargestellt und der Begriff des Situationsbewusstseins erläutert. Als nächstes werden die Herausforderungen der Mensch-Roboter Kommunikation geschildert und ein Überblick über die verschiedenen Ansätze gegeben. Der letzte Abschnitt beschreibt die Bibliotheken und Softwarepakete, welche bei der Erstellung der Applikation genutzt wurden.

---

### 2.1 Augmented Reality - Grundprinzip und Überblick

---

Unter **Augmented Reality** (AR, dt. *Erweiterte Realität*), versteht man die Verbindung der realen und virtuellen Welt. Die Realität wird kontinuierlich mit zusätzlichen virtuellen Informationen, welche an die aktuelle Position des Betrachters angepasst sind, erweitert [14] [15]. Augmented Reality wird auch durch die Verbindung der folgenden Eigenschaften definiert [16]:

- i Verbindung der Realität mit virtuellen Inhalten - Die menschliche Wahrnehmung wird um computergenerierte Inhalte erweitert.
- ii Interaktion in Echtzeit - Die Darstellung der virtuellen Inhalte wird unmittelbar auf die Veränderung der Betrachterposition angepasst.
- iii Räumliche Aufnahme - Virtuelle Objekte stehen in Bezug zur realen Umgebung.

Jedes AR-System besteht aus den folgenden Elementen: Videoaufnahme, Tracking, Registrierung, Darstellung und Ausgabe [14].

Um die reale Umgebung des Anwenders mit virtuellen Informationen anzureichern, wird diese mit einer Kamera (z.B. Webcam, Tablet- oder Industriekamera) aufgenommen. Das **Aufnahmegerät** kann dabei an ein beliebiges (statisches oder bewegliches) Objekt befestigt werden. In dem Videostream werden die Objekte positionsgerecht um die virtuellen Informationen erweitert. Um diesen Schritt erfolgreich realisieren zu können, muss die Position (Lage und Orientierung) des Betrachters (beziehungsweise Aufnahmegeräts) bezüglich des Objekts, welches man mit virtuellen Informationen erweitern will, bekannt sein. Dazu werden sogenannte Trackingverfahren benutzt.

Unter **Tracking** versteht man das Verfolgen von Objekten und das Ermitteln ihrer Position und Orientierung (Rotation). Die Trackingmethoden können in zwei Hauptkategorien aufgeteilt werden: sensorische und optische Anwendungen. Die sensorbasierten Trackingverfahren können auf mechanischen, optischen, magnetischen oder akustischen Sensoren basieren [17] [18].

---

Während des *Akustischen Trackings* werden Ultraschallwellen mit einem Mikrophon aufgenommen und zur Entfernungsberechnung genutzt. Dies ist nur für kleine Umgebungen ohne zusätzliche Geräuschquellen geeignet. *Elektromagnetische Tracking*-Systeme basieren auf Induktion. Um die im Raum platzierten Sensoren wird ein Magnetfeld erzeugt. Das Magnetfeld induziert einen Strom, der an dem Empfängersensoren gemessen werden kann. Aus diesen Daten kann die Position der einzelnen Sensoren bestimmt werden. Leider verfälschen metallische Gegenstände, die sich in der Umgebung befinden, die Ergebnisse der Messung, wodurch die Bestimmung der Position ungenau sein kann.

Im Gegensatz zu den sensorischen Tracking Systemen erreichen die optischen Tracker (markerbasiert, markerlos) eine höhere Genauigkeit bei der Positionsmessung. In diesen Systemen werden meistens Methoden der Bildverarbeitung verwendet, um die vordefinierten optischen Markierungen (Marker) im Videostream zu identifizieren und daraus Rückschlüsse auf die Position und den Betrachtungswinkel zu ziehen. Dies setzt voraus, dass sich zumindest eine Markierung im Sichtfeld der Kamera befindet.

Im markerlosen optischen Tracking werden charakteristische und eindeutig identifizierte Features (Merkmale) der Umgebung zur Bestimmung der Kameraposition genutzt. In einem Vorverarbeitungsschritt muss zunächst eine Merkmalskarte angefertigt werden, das heißt, in den Aufnahmen der Umgebung werden charakteristische Muster definiert. Als nächstes können diese in dem aktuellen Videostream mit einem Extraktionsalgorithmus identifiziert werden. Die aktuelle Position des Betrachters wird durch den Vergleich der Karte mit der aktuellen Aufnahme bestimmt. Die Abbildung 2.1 zeigt die Erweiterung der realen Szene mit einem 3D Automodell. Das Modell wird aufgrund von gefundenen Features in dem Bild positioniert.

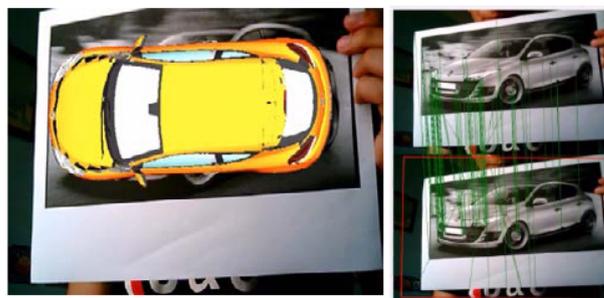


Abbildung 2.1: Erweiterung der realen Szene mit einem virtuellem Auto. Die in der Umgebung identifizierten Merkmale (rechts unten) werden mit einer Merkmalskarte (rechts oben) verglichen. **Quelle:** [19]

Das markerbasierte Tracking beruht auf der Zuordnung von bekannten Umgebungsmarkierungen auf den zu dem im Bild gefundenen Merkmalen. Diese Markierungen (Marker) haben meist eine vordefinierte Größe und Form. Wie die Abbildung 2.2 zeigt, sind die meisten Marker quadratisch und mit einem schwarzem Rahmen versehen. Sie können untereinander durch den Vergleich des inneren Musters eindeutig identifiziert werden. Die Kameraposition kann anhand von den *2D/3D* Korrespondenzen bestimmt werden.

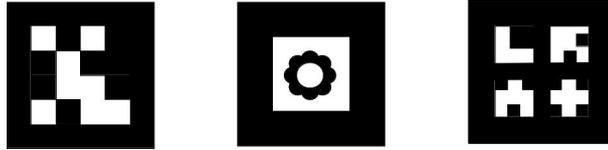


Abbildung 2.2: Beispielmarker genutzt bei optisch-markerbasierten Trackingverfahren. Die am meisten verwendeten Marker sind quadratisch und mit einem schwarzem Rahmen versehen

Ist die Kameraposition bekannt, so können die virtuellen Objekte in die reale Umgebung **registriert** werden. D.h. die virtuellen Inhalte müssen an die realen Objekte angepasst werden, so dass der Anschein entsteht, dass die beiden Welten parallel existieren und sich ergänzen. Basierend auf der Transformation zwischen dem Betrachter (Kamera) und einem Trackingobjekt wird die Perspektive für die virtuellen Inhalte bestimmt, so dass der Mensch unabhängig von seinem Standpunkt den Eindruck hat, dass alle Objekte sich am selben Ort befinden (mit Ausnahme von animierten Objekten). Die Abbildung 2.3 zeigt eine Registrierung der virtuellen Objekte in einer realen Industrieumgebung. Die virtuellen Inhalte werden direkt an der Maschine eingeblendet.



Abbildung 2.3: Registrierung der Objekte. Die reelle Umgebung wird um computergenerierte Inhalte erweitert. Die virtuellen Inhalte werden direkt an der Maschine eingeblendet **Quelle:** [20]

Der letzte Schritt eines Augmented Reality Systems bildet die Ausgabe. Sie kann mit Hilfe von Monitoren, mobilen Geräten (Tablet-Computer, Smartphone) oder einer Datenbrille erfolgen.

Insgesamt gibt es zwei Möglichkeiten die Realität zur erweitern: optisch- oder videobasiertes See-Through [16]. Die optische Ausgabe erlaubt dem Benutzer die direkte Wahrnehmung der realen Umgebung. Daraus folgt, dass auf einem transparenten Display (z.B. Datenbrille) nur die virtuellen Inhalte dargestellt werden, wie in der Abbildung 2.4 zu sehen ist. Im Gegensatz dazu benötigt Video-See-Through die Aufnahme der realen Umgebung mit einem Aufnahmegerät. Der Betrachter sieht dann die aufgenommene Szene, die um die virtuellen Inhalte erweitert wird. D.h. Realität wird auf einem Video mit dem virtuellen (Teil-)Bild überlagert.



Abbildung 2.4: See Through mit einer Datenbrille (Mini Augmented Vision). Informationen werden direkt ins Sichtfeld eines Autofahrers eingeblendet. **Quelle:** [1]

---

## 2.2 Situationbewusstsein

---

Heutzutage werden die Aufgaben, die der Mensch in der industriellen Umgebung lösen muss, immer komplexer. Vor der Einführung von verschiedenen Automatisierungskonzepten in den Industrieanlagen, wurden die Prozesse häufig manuell vom Arbeitern gesteuert. Roboter waren rein mechanische begrenzt steuerbare Werkzeuge, die über einen Hebel und Knöpfe bedienbar waren. Der Arbeiter hatte volle Kontrolle über das System. In modernen Produktionswerkstätten kommuniziert der Arbeiter mit der Maschine meistens über eine Benutzerschnittstelle, die der Überwachung und Steuerung von komplexen Produktionsprozessen dient und neue Herausforderungen mit sich bringt. Während seiner Tätigkeit muss der Mitarbeiter gleichzeitig die Umgebung, sowie verschiedene Prozessgrößen (z.B. Druck, Geschwindigkeit der Maschine, Wiederholungsrate) wahrnehmen und dafür sorgen, dass der Produktionsprozess korrekt ausgeführt wird. Das heißt der Arbeiter muss regelmäßig die Veränderungen in seiner Umgebung und im Prozess erkennen, verstehen und die Entwicklung in naher Zukunft vorhersagen. Dieser Zustand wird **Situationbewusstsein** genannt [21].

Der Begriff **Situationbewusstsein** oder **-Verständnis** (engl. *Situation Awareness*, SA) beschreibt allgemein die Herausforderungen, die in verschiedenen Situationen auf eine Person treffen. Hierzu gehört die Wahrnehmung (engl. *perception*) von Elementen in der Umgebung, das Verständnis für deren Bedeutung (engl. *comprehension*) und die Vorhersage des Zustands in der näheren Zukunft (engl. *projection*) [12]. Diese drei aufeinander aufbauenden Stufen, die in der Abbildung 2.5 dargestellt wurden, repräsentieren verschiedenen Ebenen der Informationsverarbeitung und werden im folgenden genauer beschrieben.

Als erstes werden die Umgebungsfaktoren innerhalb eines dynamischen Systems wahrgenommen. Nimmt die Person keine für die Situation relevanten Reize wahr, so kann ein inkorrektes Bild des aktuellen Geschehens entstehen.

Auf der zweiten Stufe werden wahrgenommene Daten verarbeitet (sortiert, interpretiert oder gespeichert), so dass Objekte und Ereignisse an Bedeutung gewinnen. Einzelne Größen können mit anderen kombiniert und als eine Einheit betrachtet werden.

Auf der dritten Stufe wird die dynamische Situationsentwicklung in der näheren Zukunft prognostiziert. Dieser Schritt ermöglicht dem Menschen für die jetzige Lage angemessen zu

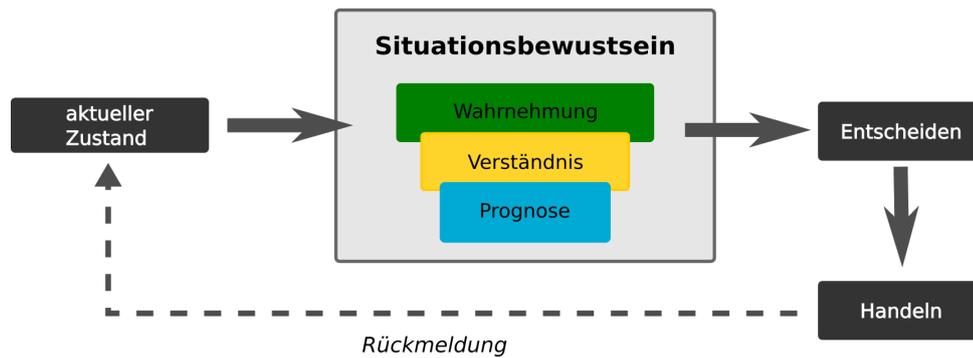


Abbildung 2.5: Situationsbewusstsein setzt sich aus drei Ebenen zusammen: Wahrnehmung von Elementen in der Umgebung, Verständnis für deren Bedeutung und Prognose des Zustands. Diese haben Einfluss auf die Sicht auf die aktuelle Situation, die getroffenen Entscheidungen und das Handeln der Menschen. **Quelle:** Modifiziert nach [12]

agieren. Dabei fließen die Entscheidungsprozesse und das Handeln nicht in das Situationsbewusstseinsmodell ein. Bei jeder Entscheidung, die getroffen wird, und jeder Handlung, die ausgeführt wird, erfolgt eine Aktualisierung des Modells.

Das Erreichen von verschiedenen Stufen der SA wird von mehreren Faktoren beeinflusst. Endsley [12] differenziert zwischen dem System und individuellen Einflussfaktoren.

System-Einflussfaktoren sind durch die Aufgabe und das System definiert. Der aktuelle Zustand des Systems, sowie seine Komplexität haben Einfluss auf die Art und Weise, wie die einzelnen Reize der Umgebung wahrgenommen werden. Stress und Übermüdung können zu geringerer Perzeption (Aufnahmevermögen) führen.

Individuelle Einflussfaktoren sind durch eigene Fähigkeiten und Erfahrungen bedingt. Eine Person trifft eine Prognose über den Zustand des Systems anhand des persönlichen Wissensstandes. Sie passt ihre Aufmerksamkeit basierend auf den Kenntnissen über die Arbeitsumgebung, vorhandene Ressourcen oder Leistungsfähigkeit, auf die ihrer Meinung nach relevanten Daten an. Die Wahrnehmung der verbliebenen Elemente in der Umgebung sinkt in Folge dessen. Die Tabellen 2.1 und 2.2 zeigen jeweils ein Situationsbewusstseinsmodell einer Person, die mit dem Roboter arbeitet. Wobei die erste Person den Roboter überwacht, während zweite mit ihm arbeitet und kein Wissen über sein inneren Zustand hat. Sie benötigen verschiedene Faktoren und Fähigkeiten um den Zustand des Roboters zu bewerten.

Da das Gedächtnis des Menschen und seine Aufmerksamkeit begrenzt sind, werden die Daten nach ihrer (subjektiven) Relevanz gewichtet. Werden die wichtigen Information nicht gesehen, oder sind nicht vorhanden, kann es zu Fehlern kommen. Um dieses Risiko zu minimieren ist die Art und Weise, sowie die Menge an dargestellten Daten wichtig. Mit einer geeigneten Gestaltung einer Mensch-Maschine-Schnittstelle kann das Situationsbewusstsein erheblich verbessert werden. Eine grundlegende Frage ist dabei welche Metrik quantitative Aussagen über das SA erlaubt.

<b>Stufe 1: Wahrnehmung</b>	<b>Stufe 2: Verständnis</b>	<b>Stufe 3: Prognose</b>
<ul style="list-style-type: none"> <li>• Geschwindigkeit</li> <li>• Genauigkeit</li> <li>• aktuelle Gelenkposition</li> </ul>	<ul style="list-style-type: none"> <li>• Relevanz einzelnen Größen</li> <li>• Funktionsweise des Roboters</li> <li>• Ergebnis einer Prozessänderung</li> <li>• Kenntnis über das aktuelle und nächste Prozesselement</li> </ul>	<ul style="list-style-type: none"> <li>• Vorhersehen der Auswirkung auf die Roboterausführung nach eigener Handlung (z.B. Kollision)</li> </ul>

Tabelle 2.1: Situationsbewusstseinsmodell für den Arbeiter im Kontrollraum einer Roboterzelle. Er kennt nur den inneren Zustand des Roboters.

<b>Stufe 1: Wahrnehmung</b>	<b>Stufe 2: Verständnis</b>	<b>Stufe 3: Prognose</b>
<ul style="list-style-type: none"> <li>• Beobachtete Änderung der Geschwindigkeit</li> <li>• physische Bewegung</li> <li>• akustische Veränderungen</li> </ul>	<ul style="list-style-type: none"> <li>• Kenntnis über die Relevanz und das Resultat der Änderungen</li> </ul>	<ul style="list-style-type: none"> <li>• Vorhersehen der Auswirkung von Änderungen und/oder eigener Handlung (z.B. Kollision)</li> </ul>

Tabelle 2.2: Situationsbewusstseinsmodell für den Arbeiter, der mit dem Roboter zusammenarbeitet und keine Kenntnis über dessen inneren Zustand hat.

Die Evaluationstechniken des Situationsverständnisses lassen sich in zwei Verfahren aufteilen: direkte und indirekte [22]. Über indirekte Messmethoden können Situationserfassung oder Situationsbewusstsein durch Leistungs- und Verhaltensmaße erfasst werden. Um die zur Evaluation benötigten Daten zu erheben, kann der Benutzer gebeten werden laut zu denken, oder alternativ seine Blickrichtung registriert werden. Durch die Analyse dieser Daten werden Rückschlüsse auf das SA gezogen.

Bei direkten Verfahren wird der Benutzer nach seinen subjektiven Erfahrungen gefragt. Diese bilden eine Basis zur Bestimmung des Situationsbewusstseins. Hierbei wird zwischen subjektiven und objektiven Methoden unterschieden. Zur subjektiven Messung wird eine Ratingskala angefertigt, deren Nutzung durch die persönliche Einschätzung des Benutzers

---

zu Grunde liegt. Objektive Messmethoden sehen die Durchführung eines „Wissenstests“ vor. Der Benutzer wird dabei gebeten an einer Befragung teilzunehmen und Fragen zu der von ihm durchzuführenden Aufgabe zu beantworten. Dabei kann die Befragung während oder nach dem Lösen der Aufgabe stattfinden.

---

### 2.3 Ergonomische Anwendungsgestaltung

---

Mit einer geeigneten Struktur des Mensch-Maschine Dialogs (Benutzeroberfläche) kann das maximale Situationsbewusstsein erreicht werden. Dazu sollten während der Schnittstellenplanung die Elemente der menschlichen Informationsverarbeitung (z.B. Wahrnehmung, Kommunikation, Kognition), sowie der Zweck und Merkmale der Aufgabe, welche die graphische Benutzeroberfläche haben soll, berücksichtigt werden.

Allgemeine Richtlinien der Gestaltung der Mensch-Maschine-Schnittstelle sind in DIN ISO 9241-110 [23], sowie in Gestaltgesetzen [24] [25] zusammengefasst. Alle Vorgaben sind voneinander abhängig. Dennoch muss während der Designphase entschieden werden, ob die Erfüllung dieser Leitsätze größere Vorteile (für den Anwender) mit sich bringt. Die Gestaltungsregeln können in den folgenden Punkten zusammengefasst werden:

- i Aufgabenangemessenheit - Effizientes Erledigen der Aufgabe sollte unterstützt werden.
  - Nur Lösungsrelevante Informationen sollten dargestellt werden
  - Hilfs-Informationen sollten dem Benutzer zur Verfügung gestellt werden
- ii Steuerbarkeit - Geschwindigkeit des Dialogablaufs soll durch den Benutzer beeinflussbar sein.
  - Der Benutzer sollte auf die Menge der ihm gezeigten Daten Einfluss haben.
- iii Erwartungskonformität - Die graphische Gestaltung des Dialogs soll stimmig sein.
  - Informationen sollen auf die gleiche Art und Weise dargestellt werden.
  - Vergleichbare Aufgaben sollten ähnlich gelöst werden können.
  - Informationen, die miteinander verbunden sind, sollten räumlich gruppiert werden.
  - Elemente des Dialogs sollten symmetrisch angeordnet werden.
- iv Fehlertoleranz - Unvorhergesehene Eingaben sollten nicht zu Abstürzen der Software führen.
- v Individualisierbarkeit - Die Inhalte des Dialogs können individuell an den Benutzer angepasst werden.
- vi Lernförderlichkeit - Der Benutzer wird beim Erlernen des Dialogs unterstützt.

---

Üblicherweise basieren Benutzerschnittstellen auf Text- und Grafikstrukturen. Soll die Kommunikation zwischen Mensch und Maschine zusätzlich durch verschiedene dynamische Medien (Video, Livestream, Animationen) unterstützt werden, so wird von einer **Multimedia Anwendung** gesprochen. Um die Multimediaschnittstelle effizient und ergonomisch zu gestalten sollten neben den allgemeinen Richtlinien der Dialoggestaltung, die speziellen Eigenschaften berücksichtigt werden [26] [27]:

- i Verwendbarkeit als Kommunikationsziel - Die verwendeten Medien sollen so gestaltet werden, dass sie die Informationen angemessen zur Aufgabe und den Informationsbedürfnissen des Benutzers übermitteln.
- ii Verwendbarkeit für Wahrnehmung und Verständnis - Die zu übermittelnden Daten sollen so strukturiert werden, dass sie leicht zu erfassen und zu verstehen sind. Es muss darauf geachtet werden, dass der Benutzer nicht durch zu viele Informationen überlastet wird. Zusätzlich sollte die Wahrnehmung der Daten nicht durch andere Tätigkeiten (z.B. Orientierungs- und Navigationstätigkeiten) beeinträchtigt werden.
- iii Verwendbarkeit für die Exploration - Durch eine gute Strukturierung der Informationen soll der Benutzer leicht die Multimedia-Applikation erkunden können.
- iv Verwendbarkeit für die Benutzungsmotivation - die Anwendung soll angemessen zur Aufgabe so gestaltet werden, dass sie das Interesse des Benutzers weckt und zur Interaktion einlädt.

Einen großen Einfluss auf die Ergonomie des Mensch-Maschine Dialogs hat die visuelle Informationsgestaltung (z.B. Gebrauch von Farben, Effekten und Schriften). Durch eine geeignete graphische Gestaltung kann die Aufmerksamkeit des Anwenders gezielt gelenkt werden. Die visuellen Effekte sollten den Nutzer bei einer schnellen Entscheidung über die Wichtigkeit der Informationen unterstützen, z.B. Hinweise auf Fehler werden auf einem roten Hintergrund dargestellt. Zusätzlich sollten Farben benutzt werden, um den Benutzer die Orientierung zu erleichtern. Allgemein sollten gesättigte Farben nicht als Hintergrundfarben verwendet werden. Gleichzeitig muss auf einen ausreichenden Farbkontrast zwischen den einzelnen Elementen und Flächen geachtet werden. Falsch gewählte Farben können von wichtigen Informationen ablenken. Um die Aufmerksamkeit des Benutzers auf Objekte in der graphischen Oberfläche zu lenken, können zusätzliche graphische Effekte, wie Blinken, Schriftarttribute oder Veränderungen der Helligkeit genutzt werden.

---

## 2.4 Roboter-Mensch Interaktion - Stand der Forschung

---

Während der Arbeit mit dem Roboter muss dem Menschen der Zustand der Maschine bewusst sein. Die Interaktion des Roboters mit dem Arbeiter dient damit der Übermittlung von Informationen über den Zustand des Systems. Durch Auswahl einer für die Situation geeigneten Kommunikationsschnittstelle, kann die Komplexität des System abstrahiert und das Verständnis für die Intention des Roboters verbessert werden.

Entsprechend dem Prinzip des Situationsbewusstseins soll der Benutzer ein besseres Gefühl dafür bekommen, was der Roboter gerade macht oder was er als nächstes tun wird. Dabei sollten die verschiedenen Daten über den Roboter dem Menschen mit Hilfe von einem oder mehreren Kommunikationskanälen übermittelt werden. Zu Beachten ist, dass eine größere Anzahl an Daten nicht den Informationen, die für das Verstehen der aktuellen Situation notwendig sind, entsprechen muss (vergleiche 2.6). Die zusätzlichen Daten können dazu führen, dass der Zustand des Systems unübersichtlich wird und das Verständnis für die aktuelle Situation stark sinkt.

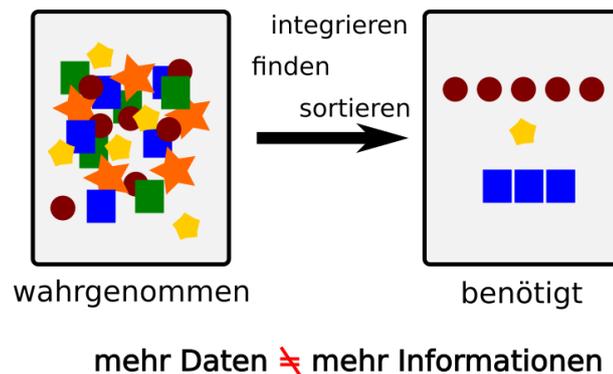


Abbildung 2.6: Informationsloch. Nicht alle wahrgenommenen Daten werden für das Verständnis der Situation benötigt. **Quelle:** Modifiziert nach [28]

Die Kommunikationskanäle basieren auf menschlichen Sinnen wie Sehen, Hören und Tasten [29]. Daher kann die Kommunikation zwischen dem Roboter und Menschen zum Beispiel über eine Projektion in die Umgebung oder einer Darstellung von Daten auf einem Display erfolgen. Zusätzlich können Roboterdaten an den Menschen mittels akustischen Signale oder Bewegungen des Roboters, übermittelt werden.

---

#### 2.4.1 Kommunikation durch Bewegung und Akustik

---

In [30] beschreibt Kobayashi eine bewegungsbasierte Kommunikation des Roboters mit dem Menschen. Der Ansatz ist als Lösung von sog. Hindernis-Entfernen-Aufgaben, welche die Interaktion des Menschen mit dem Roboter voraussetzen, vorgestellt worden. D.h. Trifft der Roboter während der Ausführung einer Aufgabe auf ein Objekt, welches von ihm nicht beseitigt werden kann, kann er durch Bewegung die Aufmerksamkeit des Menschen auf sich ziehen und so um Hilfe bitten. Nach der Entfernung des Hindernisses, setzt die Maschine ihre Arbeit fort.

Die Kommunikation basiert auf einer Vor- und Rückbewegung des Roboters, welches in der Abbildung 2.7 dargestellt wurde. Entdeckt der Roboter eine Barriere, z.B. einen Tisch, fährt er kurzzeitig vier mal hin und zurück. Diese Bewegung ist universell und dadurch für den Menschen einfach zu verstehen.

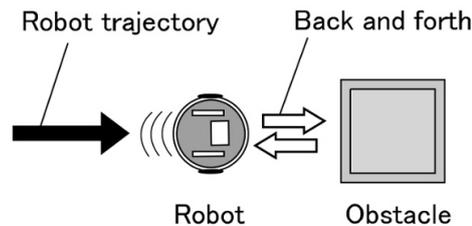


Abbildung 2.7: Bewegungs-basierte Kommunikation. Der Roboter bewegt sich Hin und Zurück vor einem Hindernis und wartet auf einer Reaktion des Menschen. **Quelle:** [31]

Die Experimenten ergaben, dass die Intention des Roboters, welche mit Hilfe von Bewegung abgebildet worden ist, für den Menschen einfacher, als andere nonverbale Kommunikationskanäle (z.B. Kommunikation über LED) zu verstehen ist.

Bewegungs-basierte Kommunikation ist flexibel, jedoch können durch die Bewegung nicht alle Roboterzustände abgebildet werden. Der Roboter kann durch eine auffällige Bewegung, die Aufmerksamkeit des Menschen auf sich ziehen, allerdings muss der Benutzer Kenntnis über die Bedeutung der Bewegung haben. Diese Art der Kommunikation setzt voraus, dass der Arbeiter den Roboter die ganze Zeit beobachtet, was in der betrieblichen Praxis schwierig zu realisieren ist.

Komatose hat in [32] untersucht, welchen Einfluss akustische Signale in der Mensch-Maschine Kommunikation haben. Für diese Studie wurden den 23 Probanden verschiedene Laute abgespielt. Für das Experiment sind 44 akustische Zeichen in elf Grundfrequenzen, die linear ab- oder aufsteigend waren, mit vier verschiedenen zeitlichen Dauern verwendet worden. Nachdem alle Signale durch einen Probanden abgehört wurden, sind Fragen zur subjektiven Bedeutung des Signals gestellt worden. Dabei sind drei Kategorien zur Bedeutung des Signals definiert worden:

- i negativ - Widerspruch, Zweifel oder Überraschung
- ii neutral - Zögern, Sprachlos
- iii positiv - Zustimmung, Akzeptanz

Die Experimente haben gezeigt, dass einfache akustische Signale mit verschiedenen Intensitäten und verschiedener Dauer als eine Art von Informationen, welche die Maschine dem Menschen übermitteln möchte, verstanden werden können. Steigende Töne, unabhängig von ihrer Dauer, werden als Widerspruch verstanden. Flache, länger andauernde Signale waren als Zögerung interpretiert worden. Zustimmung kann durch kurze, absteigende Laute ausgedrückt werden.

Akustische Interaktion ist effektiv und universell, trotzdem ist sie beschränkt. Zwar kann die Aufmerksamkeit des Benutzers mit Hilfe von verschiedenen Signalen auf die Maschine gelenkt werden, trotzdem weiß der Mensch nicht was passiert ist. Er bekommt keine Informationen über die Position oder Art des Fehlers im System.

---

## 2.4.2 Visuelle Kommunikation

---

Das Fraunhofer-Institut für Fabrikbetrieb und -automatisierung IFF hat eine Anwendung zur optischen Arbeitsraumüberwachung und Roboterzustandsdarstellung [33] vorgestellt. Dieses projektionsbasierte Überwachungssystem, welches in der Abbildung 2.8 zu sehen ist, verbindet Lichtprojektion mitameratechnik. In vier Ecken einer Arbeitszelle sind Kameras montiert worden. Der durch den Projektor gesendeter Lichtstrahl wird mit Hilfe eines Spiegels reflektiert und in die Umgebung (z.B. Boden) projiziert. Eine Unterbrechung des Projektionsmusters (Zum Beispiel verursacht von einem Menschen, dazu siehe Abbildung 2.9) führt zur Verringerung der Robotergeschwindigkeit oder dessen Anhalten, wodurch die Sicherheit des Menschen in der Nähe des Roboters gewährleistet wird.

Abhängig von der aktuellen Position der Robotergelenke und dessen Geschwindigkeit wird ein dynamischer Sicherheitsbereich visualisiert. Zusätzlich werden Zielposition, die Richtung der Bewegung sowie ein Transferbereich in die Umgebung projiziert.

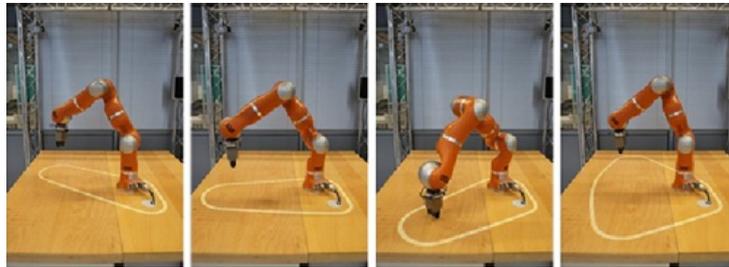


Abbildung 2.8: Arbeitsraumüberwachung mit einem kamera- und projektionsbasiertem Sensorsystem. Der auf dem Arbeitstisch projizierte Bereich wird mit jeder Bewegung des Roboters aktualisiert. **Quelle:** [34]



Abbildung 2.9: Arbeitsraumüberwachung. Unterbrechung des Lichtstrahls führt zum Anhalten des Roboters. **Quelle:** [35]

Durch die Projektion der Informationen in die Umgebung kann der Mensch direkt sehen, was der Roboter macht. Der Anwender bekommt ein besseres Gefühl für den aktuellen

Roboterzustand, er kann seine Intention besser abschätzen und die nächsten Bewegungen des Roboters vorhersagen, d.h. sein eigenes Situationsbewusstsein wird verbessert. Jedoch macht die benötigte Kameravorrichtung das System nicht flexibel einsetzbar. Der Benutzer bekommt keine zusätzlichen Informationen über den inneren Zustand des Roboters (z.B. Fehlermeldungen), wodurch Unsicherheiten entstehen können.

Die Firma Rethink Robotics hat im Jahr 2012 den zweiarmigen Baxter Roboter [36] vorgestellt (vergleiche Abbildung 2.10). Dieser wird über geführtes Teach-In programmiert. D.h. der Arbeiter führt den Roboterarm und zeigt ihm, welche Bewegung er als nächstes ausführen soll.

Baxter gibt dem Menschen eine visuelle Rückmeldung über seinen aktuellen Zustand und den von ihm ausgeführten Prozess. Er kommuniziert mit Hilfe von Emojis, die auf einem fest montiertem Display zu sehen sind. Die aktuelle Bewegungsrichtung der Arme wird z.B. über in die entsprechende Richtung blickende Augen angedeutet. Die Abbildung 2.11 zeigt eine Auswahl der darstellbaren Roboterzuständen. So zum Beispiel bedeuten geschlossene Augen, dass der Roboter in den Bereitschaftszustand gewechselt hat.



Abbildung 2.10: Baxter - zweiarmiger Roboter mit einem festmontiertem Display. **Quelle:** [2]

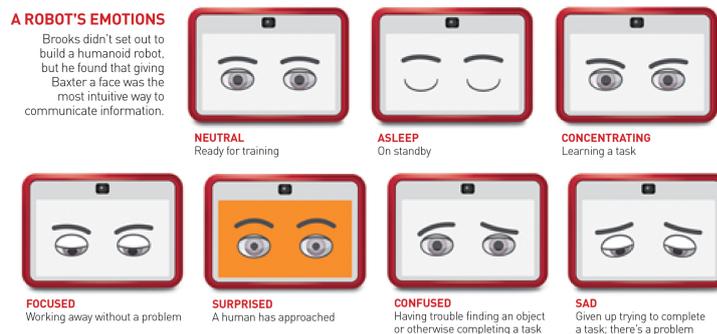


Abbildung 2.11: Baxter kommuniziert mit dem Menschen mit Hilfe von Bildzeichen. Geschlossene Augen bedeuten, dass der Roboter in den Bereitschaftszustand gewechselt hat. **Quelle:** [36]

---

Diese Art der Vermittlung von Informationen über den Zustand des Roboters ist intuitiv, aber nicht eindeutig. Die verschiedenen Emojis lassen Freiraum für Interpretationen und können nicht alle Prozesszustände (z.B. Bewegungsbahn, Arbeitsraum) des Roboters darstellen.

In [37] und [38] ist ein Ansatz der Roboter-Mensch Kommunikation für mobile Industrieroboter vorgestellt worden. Auf einem Optoma ML 750 sind ein Lichtprojektor und drei Lasersensoren (zwei vorne und einer hinten) montiert worden. Der Roboter transportiert Teile von einer Arbeitsstation zur Nächsten. Er teilt sich seinen Arbeitsbereich mit dem Arbeiter. Um dem Menschen die Intention der Maschine zu übermitteln, wird ein Spatial Augmented Reality Ansatz [39] benutzt. Das heißt es wird mit Hilfe eines Projektors eine weiße Linie projiziert, welche einer demnächst gefahrenen Bahn entspricht. Zusätzlich sind zwei Sicherheitsbereiche, wie in der Abbildung 2.12 zu sehen ist, vor dem Roboter abgebildet worden. Tritt der Mensch in den roten Bereich, so wird der Roboter automatisch gestoppt. Befindet sich der Arbeiter hingegen in dem grünen Areal, wird die Geschwindigkeit des Roboters lediglich reduziert.

Die durchgeführten Benutzerstudien haben die Annahme, dass der Mensch dem Roboter mehr vertraut, falls er Informationen über dessen Vorhaben besitzt, bestätigt. Einfache Einblendung der Trajektorie und der beiden Sicherheitsbereichen um den mobilen Roboter haben gereicht um die Mensch-Roboter Kommunikation zu verbessern.



Abbildung 2.12: Optima ML 750 - Mobiler Industrietransportroboter. Ein auf dem Roboter montierter Projektor, visualisiert den Sicherheitsbereich und seine Trajektorie  
**Quelle:** [38]

Dieser Ansatz ist viel versprechend, jedoch nur begrenzt in der Industrierobotik anwendbar. Bei der Bewegung eines Roboterarms würde die Projektion oft verdeckt werden. Bei schnellen Bewegungen des Roboters ist der auf der Boden projizierte Trajektorienabschnitt vermutlich nicht ausreichend, um dem Benutzer die Unsicherheiten zu nehmen. Zusätzlich stehen dem Anwender keine Informationen, über den allgemeinen Zustand des Roboters, zur Verfügung.

#### **Zusammenfassend:**

In der Literatur existieren verschiedenen Ansätze, welche die Kommunikation zwischen

dem Roboter und dem Menschen intuitiver gestalten. Jedoch sind nicht alle in der betrieblichen Praxis einfach einsetzbar. Oft werden mehrere Sensoren und zusätzliche Vorrichtungen benötigt. Die meisten Ansätze geben dem Benutzer keine Rückmeldung über die aktuell aufgetretenen Fehler. Er wird nicht über den gesamten Ablauf des Prozessplans oder den allgemeinen Zustand des Roboters (Instandhaltung) informiert.

---

## 2.5 Bibliotheken und Softwarepakete

---

### 2.5.1 Robot Operating System

---

**Robot Operating System** (*Ros*) [40] [41] ist eine open-source Middleware Plattform, die verschiedene Bibliotheken und Werkzeuge zur Roboterprogrammierung bereitstellt. Das Grundkonzept basiert auf Knoten, Nachrichten, Services und Topics. In der Abbildung 2.13 ist der Kommunikationsprinzip zwischen zwei Rosknoten dargestellt worden, dessen Komponenten werden im Folgenden genauer beschreiben.

**Knoten** sind Prozessmodule, die die Aufgaben ausführen. In einem System können mehrere Knoten gestartet werden, die bestimmte Funktionen in der *Ros*-Umgebung abkapseln und miteinander mittels **Nachrichten** kommunizieren. Diese werden aus vordefinierten Datenstrukturen, wie primitive Typen (Integer, Double, Float, usw.) oder Arrays von primitiven Typen bzw. anderen Nachrichten gebildet. Zum Austausch von Meldungen werden Topics und Services benutzt.

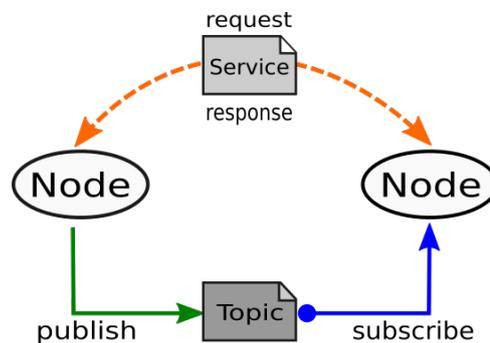


Abbildung 2.13: *Ros*-Kommunikationskonzept. Zwei Knoten kommunizieren miteinander über Topics und Services. Topics können, im Gegensatz zu Services, durch mehrere Rosknoten veröffentlicht und gelesen werden. **Quelle:** Modifiziert nach [40]

- i Topics - sind einfache Variablen, die sich wie ein virtuelles Bus-System verhalten und asynchron arbeiten. Sie werden zur unidirektionalen Kommunikation benutzt. Das bedeutet eine zum Topic gehörende Nachricht wird permanent von einem oder mehreren Senderknoten veröffentlicht und kann durch einen oder mehrere Empfängerknoten gelesen werden.
- ii Services - werden durch seinen Namen, so wie vordefinierte Nachrichtentypen für Anfrage und Antwort definiert. Sie arbeiten synchron. Das heisst ein Service muss

---

von einem Knoten mit einer Sendernachricht aufgerufen werden und bekommt eine Antwortnachricht zurück. Im Gegensatz zu einem Topic kann ein Service nur von einem Knoten gesendet werden.

### Unified Robot Description Format

Unified Robot Description Format [42] [43] (*urdf*) ist ein von *Ros* unterstützte Beschreibungssprache von Robotermodellen, die XML-basiert ist. Es beschreibt die Abmaße der einzelnen Robotererelemente sowie deren Relation zueinander. Das heißt *urdf* bietet eine Möglichkeit den Roboter über Parameter zu beschreiben, welche die Grenzen für die Gelenke, Vorschriften der Kinematik, sowie das dynamische Verhalten des Roboters festlegen. Mit Hilfe von dieser Beschreibung kann die Relation der einzelnen Roboterframes zueinander generiert werden.

### Transform - Koordinatenverwaltung in *Ros*

Bei der Programmierung eines Roboters ist es oft notwendig die richtige Anordnung der Roboterkomponenten und zusätzlichen Sensoren (wie eine Kamera) in Raum und Zeit zu kennen. Zum Beispiel: Der Roboter soll einem Objekt, dessen Position bezüglich der fest auf dem Roboter montierten Kamera angegeben wird, mit dem Endeffektor folgen. So muss bei der Positionsänderung des Objekts die Ziellage des Endeffektors neu bestimmt werden. Dazu müssen die relativen Lagen der einzelnen Roboterkomponenten bekannt sein.

*Ros* bietet zur Verwaltung dieser Informationen ein Transform (*Tf*) Paket an. *Tf* [44] speichert den Zusammenhang zwischen einzelnen Frames in einer Baumstruktur ab. Jeder Frame ist mit einem Zeitstempel versehen.

Der Benutzer braucht kein Hintergrundwissen über die Anordnung und Position der einzelnen Komponenten des Robotersystems. So wird zum Beispiel durch die Eingabe des Namens eines Ziel- bzw. Startframes eine Relation der beiden Koordinatensystemen zurückgegeben. Zusätzlich wird der Anwender bei der Transformation von Punkten und Vektoren zwischen verschiedenen Frames unterstützt.

### Rosbridge

Rosbridge [45] [46] ist ein *Ros*-Paket, welches eine abstrakte Schicht innerhalb eines Systems bildet. Es ermöglicht *Ros*-Style Kommunikation zwischen einer nicht-*Ros* Komponente, wie eine Webapplikation und dem Robotersystem, was neue Möglichkeiten in der Mensch-Roboter-Interaktion bietet.

Einzige Voraussetzung an das Klientensystem ist, dass es IP-Sockets unterstützt. Das Rosbridge Protokoll ermöglicht Zugriff auf *Ros* Strukturen, welche als JASON (JavaScript Object Notation) Objekte an die Klientapplikation weitergeleitet werden. Über diese Komponente können die Topics publiziert und abgehört werden, die Parameter gesetzt sowie Services abgerufen werden.

Um die Kommunikation zwischen dem Rosbridge und der Webapplikation zu vereinfachen, ist eine JavaScript Bibliothek **Ros.js** [47] entwickelt worden. Wie der Abbildung 2.14 entnommen werden kann, setzt sich eine Roboter-Webapplikation aus folgenden Kompo-

---

neneten zusammen: Roboter, Rossoftware, Rosbridge, ros.js und der Weboberfläche. Dabei bildet Rosbridge eine Schnittstelle zwischen einer nicht-Ros und Ros Komponente.

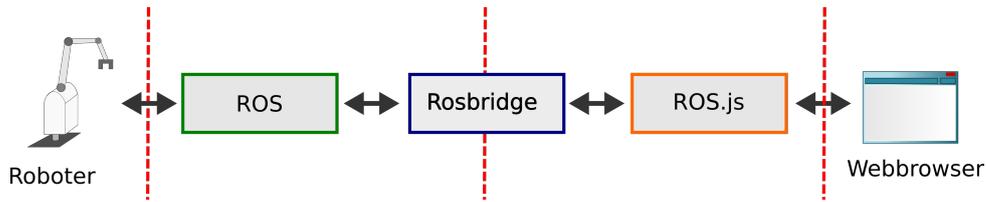


Abbildung 2.14: Rosbridge Pipeline für eine Webapplikation, welche mit einem Ros-System kommuniziert. Rosbridge bildet eine Schnittstelle zwischen einer nicht-Ros und Ros Komponente. **Quelle:** Modifiziert nach [47]

---

## 2.5.2 ARToolKit

---

ARToolKit [48] ist eine der bekanntesten Bibliotheken zur Entwicklung von Augmented Reality Anwendungen. Dieses frei verfügbare Softwarepaket ist durch Hirokazu Kato und Mark Billinghurst im Jahr 1999 zum ersten Mal vorgestellt worden [49]. Es ist in C/C++ entwickelt und wird durch alle Betriebsplattformen, wie Windows, Linux, IOS oder Android unterstützt.

Sie beinhaltet Funktionen für die robuste Markerverfolgung und Kamerakalibrierung. Mit einem Video-Modul wird zudem eine Kameraansteuerung unterstützt. Einblenden der computergenerierten Strukturen wird dank einem Gsub-Modul, welches auf OpenGL und GLUT basiert, möglich.

---

## 2.5.3 OpenCV

---

Die OpenCV (Open Source Computer Vision) [50] [51] Bibliothek beinhaltet Algorithmen die zur Bildanalyse und -verarbeitung dienen. Sie ist 1999 im Intel Labor entstanden. Im Jahr 2006 wurde die erste Version der Bibliothek vorgestellt, wobei die aktuelle Version die Nummer 3.0 (2015) hat.

OpenCv verfügt über ein C,C++, Python sowie Java Interface. Es wird von den meisten Betriebssystemen unterstützt, z.B. Windows, Linux, IOS oder Android.

Die Bibliothek bietet Unterstützung bei der Lösung von vielen Problemen der Bildverarbeitung und -erkennung. Z.B. Gesichtserkennung, Bildsegmentierung, Objekterkennung, Kantenerkennung, Gestenerkennung, Motion Tracking, Mensch-Computer-Interaktion und ist in verschiedene Module aufgeteilt. So wird in *opencv\_core* die Basisfunktionalität, d.h. Basisdatenstrukturen und Arithmetische Funktionen definiert. *opencv\_imgproc* beinhaltet die Grundfunktionen der Bildverarbeitung, wie Bildfilter, geometrische Transformationen oder Methoden struktureller Bildanalyse.

---

## 3 Konzepte und Methoden

---

### 3.1 Problemstellung und Anforderungsanalyse

---

Um die Intention eines Industrieroboters seinem menschlichem Partner zuverlässig übermitteln zu können, müssen neue Verfahren entwickelt werden.

Ein Ziel dieser Arbeit ist, neben Erstellung einer intuitiven, graphischen Benutzeroberfläche, die zusätzlichen Informationen über den Roboter mit Hilfe der AR direkt in der Umgebung darzustellen. Grundlegend ist dabei der Einsatz von einer Kamera und die Kalibrierung des Gesamtsystems, welche eine präzise Überlagerung der realen Objekte mit den virtuellen Inhalten ermöglicht.

Das zu entwickelte System soll mit einem minimalen Aufwand in einer Industrieumgebung eingesetzt werden können. Dadurch sollen bei der Systemkalibrierung keine zusätzlichen Vorrichtungen benötigt werden und die Änderung der Roboter- oder Kameraposition keine erneute Kalibrierung erfordern. Aus diesen Gründen soll ein markerbasiertes Trackingverfahren zur Bestimmung der Kameraposition verwendet werden. Des Weiteren erfordert die Darstellung der virtuellen Informationen die Kalibrierung des Gesamtsystems, das heißt sämtliche Komponenten des Systems (Roboter, Kamera, Marker) müssen in Bezug zueinander gebracht werden.

Durch das Platzieren der Marker auf dem Roboter soll die erneute Kalibrierung, bei der Veränderung der Position des Roboters oder der Kamera überflüssig und somit in Regel nur einmal pro Roboter durchgeführt werden.

Um eine robuste Visualisierung der Daten zu ermöglichen, sollen beliebig viele Marker willkürlich auf dem gewählten Roboterglied platziert werden. Beispielweise können auf einem sich um die z-Achse drehenden, quaderförmigen Roboterglied zwei Marker auf der vorderen, drei auf der hinteren Seite und jeweils einer seitlich platziert werden. Durch die Verwendung von mehreren Marker wächst die Wahrscheinlichkeit, dass bei der Bewegung des Roboterarmes zumindest ein Marker unverdeckt und somit für die Kamera sichtbar bleibt.

In den üblichen Augmented Reality Systemen wird angenommen, dass entweder die Kamera oder der Marker fest in der Umgebung positioniert sind. Durch die Verwendung einer beweglichen Kamera und einem sich bewegenden Marker auf dem Roboter, kann die Applikation in räumlich begrenzten Arbeitsbereichen eingesetzt werden. Es wird kein unbeweglicher Robotersockel oder ähnliches benötigt, um die Marker auf ihm platzieren zu können.

Durch den erarbeiteten Ansatz soll der Anwender das Robotersystem besser verstehen lernen. Jedoch kann nicht vorausgesetzt werden, dass alle Personen, die in der Nähe eines Roboters arbeiten über ein ausgeprägtes Robotikwissen verfügen. Es wird deshalb die Anforderung gestellt, dass die gewählte Visualisierung möglichst intuitiv sein soll. Das Verständnis für die einzelnen Elemente des Roboterzustands und -prozesses sollten

---

durch Bilder, Grafiken und Augmented Reality-Elemente unterstützt werden. Komplexe Roboterabläufe sollten so dargestellt werden, dass auch der unerfahrene Anwender in der Lage ist, die Intention des Roboters nachzuvollziehen, zu können und sein Zustand zu bewerten. Die zu darstellenden Roboterparameter und -variablen sollten so gewählt werden, dass sie dem Anwender helfen eine klare Sicht auf das Innere des Roboters werfen zu können.

Die Applikation soll den Benutzer während der Arbeit unterstützen. Deshalb muss bei der Entwicklung der graphischen Benutzeroberfläche die Benutzerfreundlichkeit im Vordergrund stehen, was die Berücksichtigung von Dialoggestaltungsgrundsätzen (siehe 2.3) impliziert. Die Anwendung soll auch für eine unerfahrene Person einfach bedienbar und durchschaubar sein. Die einzelnen Funktionalitäten sollten schnell auswählbar und erkennbar sein.

Zusätzlich soll ein Systemaufbau gewählt werden, der auf verschiedenen Betriebsplattformen ausführbar ist und eine effiziente und zuverlässige Informationsdarstellung ermöglicht. Das System sollte sowohl einfache Webcams mit geringerer Auflösung, als auch industrielle Kameras unterstützen. Zusätzlich soll *Ros* (siehe 2.5.1) als Schnittstelle zwischen dem Robotersystem und der Anwendung verwendet werden. Dies ermöglicht eine universelle Kommunikation zwischen den Einheiten.

---

## 3.2 Grundlagen

---

Für die Beschreibung der angewandten Konzepte folgen in nachstehenden Kapiteln zunächst einige Definitionen.

---

### 3.2.1 Zerlegung einer Transformationsmatrix

---

Eine **Transformation** bzw. eine **homogene Transformationsmatrix** [52] beschreibt die translatorischen, rotatorischen und perspektivischen Abbildungen und hat die Form:

$$T \in \mathbb{R}^{4 \times 4} = \begin{bmatrix} a_{0,0} & a_{0,1} & a_{0,2} & a_{0,3} \\ a_{1,0} & a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,0} & a_{2,1} & a_{2,2} & a_{2,3} \\ 0 & 0 & 0 & 1 \end{bmatrix} = \left[ \begin{array}{ccc|c} R & & & t \\ \hline 0 & 0 & 0 & 1 \end{array} \right]$$

mit dem Translationsvektor  $t \in \mathbb{R}^{3 \times 1}$  und der Rotationsmatrix  $R \in \mathbb{R}^{3 \times 3}$ , wobei  $R$  eine orthogonale Matrix mit  $\det(R) = 1$  ist. Es gilt für den translatorischen Teil:

$$T_t = \begin{bmatrix} 1 & 0 & 0 & t_0 \\ 0 & 1 & 0 & t_1 \\ 0 & 0 & 1 & t_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

und die rotatorische Komponente:

$$T_r = \begin{bmatrix} r_{0,0} & r_{0,1} & r_{0,2} & 0 \\ r_{1,0} & r_{1,1} & r_{1,2} & 0 \\ r_{2,0} & r_{2,1} & r_{2,2} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

**Hinweis:**

Im Allgemeinen gilt, dass das Transformierte einer Rotationsmatrix gleich der inversen Matrix ist:

$$(R)^{-1} = (R)^T,$$

jedoch kann diese Eigenschaft nicht auf homogene Transformationsmatrizen übertragen werden:

$$(T)^{-1} \neq (T)^T.$$

### 3.2.2 Frame - Relative Lage des Objekts

Unter einem **Frame** versteht man eine Beschreibung der Lage eines Gegenstandes [53]. Jedes Objekt im Arbeitsraum kann bezüglich eines externen (Weltkoordinaten) oder lokalen Koordinatensystems (d.h. relativ zu einem Objekt) beschrieben werden. Jedes Frame ist gegeben durch den Ursprung  $O$  und drei Basisvektoren  $x, y, z$ , welche die Koordinatenachsen aufspannen.

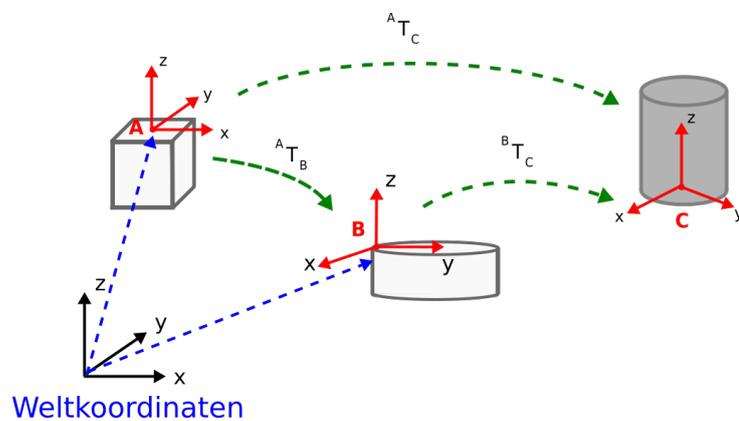


Abbildung 3.1: Framekonzept. Die Lage eines Objekt kann durch ein Koordinatensystem, welches mit einem Objekt verbunden ist, beschrieben werden. Die Position der einzelnen Objekte kann über die entsprechende Transformation (Rotation und Translation) in Relation zu einem anderem Objekt gebracht werden.

Die (relative) **Lage einer Frames** wird durch Rotation und Verschiebung bezüglich eines anderen Koordinatensystems beschrieben.

Seien Frames A,B,C wie in der Abbildung 3.1 gegeben. So beschreibt  ${}^A R_B$  die Rotation und  ${}^A t_B$  die Translation von B bezüglich A. Beide Größen bilden eine Transformationsmatrix  ${}^A T_B$ :

$${}^A T_B = \left[ \begin{array}{c|c} {}^A R_B & {}^A t_B \\ \hline 0 & 1 \end{array} \right].$$

Sind die Transformationen der Frames C bzgl. B und B bzgl A bekannt, so definiert eine Transformationsmatrix  ${}^A T_C$  die Transformationen zwischen den Systemen:

$$\begin{aligned} {}^A T_C &= {}^A T_B {}^B T_C \\ &= \left[ \begin{array}{c|c} {}^A R_B {}^B R_C & {}^A R_B {}^B t_C + {}^A t_B \\ \hline 0 & 1 \end{array} \right]. \end{aligned}$$

Die inverse Umwandlung ist wie folgt definiert:

$$\begin{aligned} {}^C T_A &= ({}^A T_C)^{-1} \\ &= \left[ \begin{array}{c|c} ({}^A R_C)^{-1} & -({}^A R_C)^{-1} {}^A t_C \\ \hline 0 & 1 \end{array} \right] \\ &= \left[ \begin{array}{c|c} ({}^A R_C)^T & -({}^A R_C)^T {}^A t_C \\ \hline 0 & 1 \end{array} \right] \end{aligned}$$

und legt die **relative Position** von C bzgl. A fest.

### 3.2.3 Kameramodell

Die Funktionsweise einer herkömmlichen Kamera kann auf das einfache Lochkameramodell zurückgeführt werden. Während einer Bild-/Videoaufnahme werden die Weltpunkte auf die Bildebene projiziert. Diese Abbildung wird durch die Kameraparameter festgelegt und kann als eine Abbildung eines dreidimensionalen Raumes auf eine zweidimensionale Ebene verstanden werden.

**Lochkamera** [55] [54] ist eine einfache Kamera, welche, wie in der Abbildung 3.2 zu sehen ist, aus einer dunklen Kammer mit einer kleinen Öffnung (Lochblende) besteht. Der Lichtstrahl von jedem Punkt des abzubildendem Raums fällt durch diese Öffnung und trifft auf die Rückwand der Kammer, auf der ein Abbild der realen Welt projiziert wird. Die Bildschärfe wird durch die Größe der Öffnung bestimmt, d.h. je größer die Lochblende, desto unschärfer das Abbild, dass auf der Rückwand der Kammer kopfüber entstanden ist.

Die Abbildungseigenschaften einer Lochkamera werden durch folgende Größen bestimmt [54]:

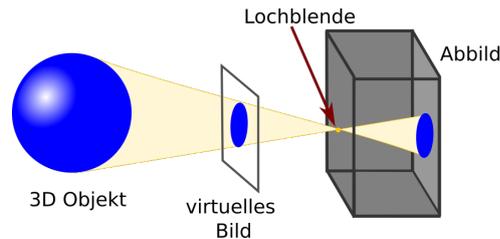


Abbildung 3.2: Eine Lochkamera besteht aus einer dunklen Kammer mit einer kleinen Öffnung. Der durch die Öffnung einfallende Lichtstrahl trifft auf die virtuelle Bildebene und die Rückwand der Kammer, wodurch eine 2D-Abbildung der realen Welt entsteht. **Quelle:** Modifiziert nach [54]

- i optisches Zentrum - stimmt mit dem Kamerakoordinatenursprung überein
  - ii optische Achse - z-Achse des Kamerakoordinatensystem, verläuft senkrecht zur Bildebene
  - iii Bildhauptpunkt - Punkt in welchem die optische Achse auf die Bildebene trifft
  - iv Brennweite - die Entfernung zwischen der Bildebene und dem optischem Zentrum
- Dessen räumliche Einordnung kann der Abbildung 3.3 entnommen werden.

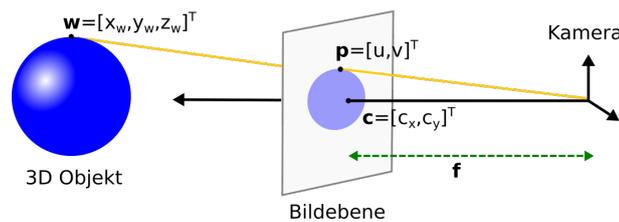


Abbildung 3.3: Kameramodell. Das optische Zentrum liegt im Ursprung des Kamerakoordinatensystems. Die virtuelle Bildebene ist auf der optischen Achse im Abstand  $f$  (Brennweite) zu Kamera positioniert. Der Punkt wo sich die Bildebene und die optische Achse treffen wird Bildhauptpunkt genannt **Quelle:** Modifiziert nach [54]

Für die Beschreibung des idealisierten Kameramodells einer herkömmlichen Kamera wird vorausgesetzt, dass die Lochblende infinitesimal klein ist. Zusätzlich wird eine virtuelle Bildebene zwischen der Lochblende und dem abzubildendem Objekt platziert, dadurch kann die Verdrehung des Abbilds vermieden werden.

Der Abbildungsprozess entspricht mathematisch einer perspektivischen Projektion des dreidimensionalen Objekts auf eine zweidimensionale Ebene und wird durch intrinsische und extrinsische Parameter fest gelegt.

Die **intrinsischen Parameter** beschreiben die innere Geometrie der Kamera. Sie legen den Zusammenhang zwischen dem 3D Kameraframe und dem 2D Bildkoordinatensystem fest und können in einer **Kameramatrix  $K$**  zusammengefasst werden:

$$K = \begin{bmatrix} f & 0 & c_x \\ 0 & f & c_y \end{bmatrix}$$

wobei  $\mathbf{f}$  die Brennweite und  $\mathbf{c} = (c_x, c_y)$  der Bildhauptpunkt sind.

Um die 3D Objektpunkte auf der Bildebene abbilden zu können, wird als erstes angenommen, dass sich die Kamera in dem gleichen Koordinatensystem, wie die abzubildenden Objektpunkte befindet. D.h. Der Ursprung der Kamera- und Weltkoordinaten stimmen überein. Die Kameraabbildung ist dann wie folgt definiert:

$$\mathbf{P}_z : \mathbb{R}^2 \times \mathbb{R} \setminus \{0\} \rightarrow \mathbb{R}^2 : \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} f \frac{x}{z} + c_x \\ f \frac{y}{z} + c_y \end{bmatrix}.$$

In Matrixschreibweise und homogenen Koordinaten ist die Position eines 3D Punktes im Bild durch das folgende Gleichungssystem bestimmt:

$$\begin{bmatrix} \mathbf{p} \\ 1 \end{bmatrix} = \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = K \begin{bmatrix} \mathbf{w} \\ 1 \end{bmatrix} = \begin{bmatrix} s_x f & 0 & x_c & 0 \\ 0 & s_y f & y_c & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}.$$

Ist das Kameraframe verschieden vom Weltkoordinatensystem, muss für alle Raumpunkte die relative Position zur Kamera bestimmt werden. Zweck dieser Transformation ist, dass die Abbildungsgleichungen in einem festem Kamerakoordinatensystem charakterisiert werden können. Die **extrinsischen (äußeren) Parameter** legen diese Transformation fest. D.h. sie beschreiben den Zusammenhang (Rotation und Translation) zwischen dem Kamera- und dem Weltframe. Sie hängen nicht von intrinsischen Parametern ab.

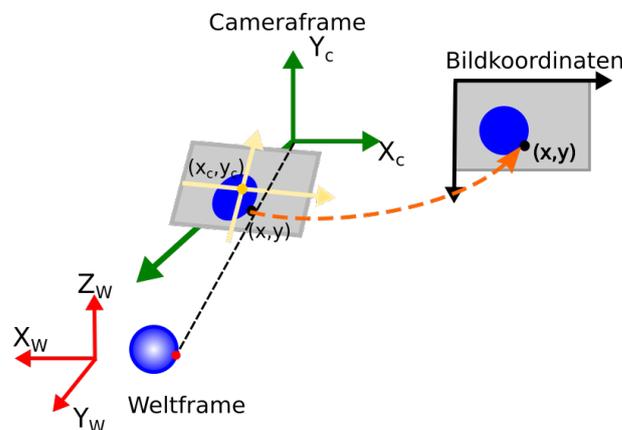


Abbildung 3.4: Transformation von Welt- über Kamera- in Pixelbildkoordinaten erfolgt mit Hilfe von einer Projektionsmatrix, die die extrinsischen und intrinsischen Eigenschaften der Kamera beschreibt.

Das komplette Kameramodell beinhaltet die extrinsischen so wie die intrinsischen Parametern und wird **Projektionsmatrix** genannt:

$$\begin{bmatrix} \mathbf{p} \\ 1 \end{bmatrix} = \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = K^W T_c \begin{bmatrix} \mathbf{w} \\ 1 \end{bmatrix} = \begin{bmatrix} s_{xf} & 0 & x_c & 0 \\ 0 & s_{yf} & y_c & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{0,0} & r_{0,1} & r_{0,2} & t_0 \\ r_{1,0} & r_{1,1} & r_{1,2} & t_1 \\ r_{2,0} & r_{2,1} & r_{2,2} & t_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}.$$

### 3.3 Roboterzustand und -prozess

Der Roboterzustand setzt sich aus verschiedenen Sensor- und Antriebssignalen zusammen, die wie in der Abbildung 3.5 zu sehen ist, während des Betriebs durch eine Kontrolleinheit überwacht und interpretiert werden. Anhand dieser Daten können direkt Roboteraktionen, wie z.B. Ausführung der Trajektorie, durchgeführt werden oder langfristige Prognosen über die zukünftigen Wartungsarbeiten getroffen werden. Diese Daten stehen dem Benutzer meistens nur als numerische Werte zur Verfügung. Um sie zu verstehen und zu interpretieren muss der Anwender oft über Hintergrundwissen verfügen.

Jedoch kann das Verständnis für den Zustand des Roboters mit Hilfe von graphischen Elementen wie Bildern, Graphen oder direkte Einblendung der entsprechenden Daten in ein Videostream gesteigert werden. Zusätzlich mindert eine sinnvolle Reduktion von dargestellten Daten den Datenoverlow (vergleiche Abbildung 2.6) und steigert, so dass Situationsbewusstsein.

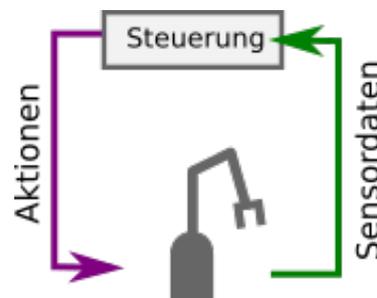


Abbildung 3.5: Roboterzustand setzt sich aus verschiedenen Sensorsignalen zusammen. Sie werden durch eine Kontrolleinheit entsprechend interpretiert und in Roboteraktionen umgewandelt.

Die Informationen über den Roboterzustand können in drei Kategorien aufgeteilt werden:

- i Statusinformationen(Pose, Geschwindigkeit) - bilden den aktuellen Zustand des Roboters ab
- ii Intention - bilden das Vorhaben, also das Prozess der Maschine ab
- iii Wartungsdaten - geben dem Benutzer die Möglichkeit, den inneren Zustand des ganzen Systems einzusehen und zu bewerten.

---

### 3.3.1 Statusinformationen

---

Statusinformationen eines Roboters beschreiben die Lage der Gelenke und dessen Geschwindigkeit. Diese Werte werden stets mit Hilfe von Robotersensoren ermittelt und stehen so grundsätzlich zu Verfügung.

#### **Pose**

Eine Pose beschreibt die Position und Orientierung eines Punktes im Raum. Konkret wird die Pose der einzelnen Gelenke, sowie des Endeffektors dargestellt.

Im Allgemeinen wird die Gelenkposition bzgl. eines festen Referenzkoordinatensystems angegeben, dazu werden die lokalen Gelenklagen entsprechend transformiert. Die Information wo sich die Robotergelenke (vor allem der Endeffektor) gerade befinden ist grundlegend für die Durchführung der Aktionen, z.B. Abfahren einer Trajektorie.

Bei Mensch-Roboter-Interaktion, kann der Benutzer meistens mit bloßem Auge die räumliche Lage der Robotergelenke bestimmen. Vorausgesetzt er weiß im Ansatz wie der Roboterarm mechanisch aufgebaut ist. Nichts desto trotz kann die Visualisierung dabei helfen die Genauigkeit, Singularitäten oder Kollisionsgefahren abzuschätzen.

Zur Steigerung des Situationsbewusstseins, wird zum einem die Position der Gelenke in einer virtuellen 3D-Umgebung visualisiert, wodurch der Benutzer die Perspektive, aus welcher er den Roboter betrachten will, selber wählen kann. Zum anderen wird die Lage der Gelenk in einer Tabelle aufgeführt.

Zur Darstellung der relativen Lage des Roboters im Bezug auf den Benutzer wird der Ansatz der Augmented Reality verwendet. Der mit der Kamera aufgenommener Live-stream wird mit der Gelenkposition überblendet. Dabei kann der Anwender durch Wechsel der eigener Position den Blickwinkel, in welchem er die Daten betrachten will, ändern.

#### **Geschwindigkeit**

Die Geschwindigkeit von jedem Gelenk des Roboters wird im allgemeinen durch die Schnelligkeit des Endeffektors festgelegt. Während des Roboterbetriebs in der Nähe des Menschen muss laut der Norm *ISO TS 15066 2016* [7] die Geschwindigkeit begrenzt werden. Mit der reduzierten Endeffektorgeschwindigkeit sinkt auch das Verletzungsrisiko. Jedoch kann die Schnelligkeit des Roboterarmes durch unterschiedliche Benutzer verschieden interpretiert und wahrgenommen werden und schon bei geringen Geschwindigkeiten Unsicherheit bei Anwendern auslösen.

Durch eine Farbkodierung des Geschwindigkeitswertes, z.B. grün für langsam, rot für schnell, wird sichergestellt, dass alle Personen die gleiche Vorstellung von der aktuellen Schnelligkeit des Roboters haben. Eine zusätzliche Darstellung in Form eines allgemein gebräuchlichem Tachometers kann das Roboter-Mensch Verständnis verbessern.

---

### 3.3.2 Intention

---

Intention beschreibt das Vorhaben eines Roboters. Dazu gehört der Prozessplan, die sich daraus ergebende Trajektorie, sowie der Sicherheitsraum. Die Vermittlung dieser Infor-

---

mationen ist von großer Bedeutung für den Menschen, da er dadurch eigenes Verhalten an den Roboter anpassen kann.

### **Prozessplan**

Hauptziel eines Industrieroboters ist das Greifen, Verpacken, Bearbeiten oder Prüfen von Objekten [56]. Der aktuelle Prozessplan legt fest, wie und wann diese Robotertätigkeiten ausgeführt werden.

Durch eine Auswahl von einzelnen Prozesselementen wird die Bewegung oder die Bewegungslogik (Warten auf ein externes Signal, Pause, Wiederholung) festgelegt. Bewegungsabläufe können in einem Teach-In (die einzelnen Punkte werden manuell angefahren) oder Offline-Verfahren (Bewegungsabläufe werden in Prozesseditor ohne Roboter festgelegt) definiert werden.

Bei einer direkten Sicht auf den Roboter ist meistens die aktuelle und darauf folgenden Aufgabe und deren Dauer nicht erkennbar. Dies kann im Benutzer Unsicherheiten auslösen, da er nicht weiß was als nächstes passieren wird. Er kann nicht beurteilen welches Prozesselement aktuell ausgeführt wird und ob dessen Ausführung abgeschlossen ist.

Eine Visualisierung des Prozesses würde das Situationsbewusstsein des Anwenders steigern. Durch die kontinuierliche Anzeige des Prozesszustands kann der Benutzer beurteilen, ob er ordnungsgemäß ausgeführt wird.

Zur Darstellung eignen sich eindimensionale Liniendiagramme oder Listen. Zusätzlich kann das aktuelle Prozesselement farblich hervorgehoben werden.

### **Sicherheitsraum und die Trajektorie**

„Eine Trajektorie (engl. trajectory) ist eine Bahn mit zeitlicher Abhängigkeit, d.h. Position, Geschwindigkeit und Beschleunigung sind in jedem Punkt der Bahn im Raum gegeben.“ [57]

Die Trajektorie ist die Bewegungsbahn des Endeffektors, die sich aus dem Prozessplan ergibt. Der aktueller Sicherheitsraum ist definiert als ein Raumbereich um den Roboter, der sich aus der auszuführenden Trajektorie und der Position der Roboterbasis ergibt.

Es sollte darauf geachtet werden, dass sich in diesem Bereich weder Personen noch Objekte befinden, die mit dem Roboter kollidieren können.

Für den Benutzer ist es schwierig ohne zusätzliche Hilfsmittel, wie Markierungen in der Umgebung des Roboters, den Sicherheitsbereich zu erkennen. Vor allem, weil er nicht unbedingt weiß, welches Prozesselement gerade oder als nächstes (und somit welche Trajektorie) ausgeführt wird. Mit der Visualisierung des Sicherheitsraumes sinkt auch das Risiko des Zusammenstoßes mit dem Roboter.

Zur Darstellung eignet sich, ähnlich wie in der Darstellung der Pose, eine Überblendung des Videostream mit den entsprechenden Daten.

---

### 3.3.3 Wartungsinformationen

---

Unter **Wartungsinformationen** versteht man die langfristigen Messungen und Faktoren, aus welchen man Rückschlüsse auf den inneren Zustand des Roboters ziehen kann. Abweichungen oder negative Trends können frühzeitig erkannt werden, wodurch eine ent-

---

sprechende Maßnahme eingeleitet werden kann. Dies führt zu kürzeren Stillständen und erhöht die Produktivität.

Eine wichtige Voraussetzung für die Wartungshilfe ist die Existenz solcher Messreihen. Zum Beispiel kann die Gelenkgeschwindigkeit, -beschleunigung oder die Genauigkeit während Ausführung einer Aufgabe mitgeloggt werden. Wodurch Abweichungen im Produktionsprozess schneller festgestellt und eventuell beseitigt werden können. Die Analyse der Betriebs- und Prozessausführungsdauer hilft den optimalen Zeitpunkt der Wartung zu bestimmen. Zusätzliche Sensoren (z.B. Temperatursensor), die den Roboter überwachen, und dessen Messungen können Hinweise auf die Ermüdung der Roboterteile liefern.

Durch die graphische Darstellung der Messdaten kann dem Benutzer ein klares Bild vom Roboterzustand übermittelt werden. Zur Darstellung von langfristig überwachten Positions- und Orientierungsgenauigkeiten können Trendkurven genutzt werden. Wobei die Steigung der Kurve der Zustandsverschlechterung des Roboters entspricht. Visualisierung der Prozessausführungszeiten oder -ausnahmen als Balkendiagramm, ermöglicht dem Benutzer einen schnellen Vergleich und Sicherstellung einer konstanten Qualität des Prozesses sowie das Eingreifen bei der Überschreitung von Fehlertoleranzen.

---

### 3.4 Kalibrierung der AR-Komponente

---

Um die realen Kamerabilder mit den virtuellen Roboterinformationen anreichern zu können, muss ein virtuelles Modell der realen Welt bestimmt werden. Bei seiner Bestimmung muss bekannt sein, an welcher Position und mit welcher Orientierung die virtuellen Objekte jeweils dargestellt werden sollen. Demnach müssen der Roboter, die Kamera und Marker in Bezug zu einander gebracht werden.

Die Abbildung 3.6 zeigt die einzelnen Schritte, die bei der Kalibrierung des Systems und Objektregistrierung benötigt werden:

- i Marker-Roboter Kalibrierung - im Bild grün markiert, es wird eine feste Beziehung zwischen dem Roboter und allen Marker bestimmt. Dazu wird eine feste Position der Kamera vorausgesetzt (siehe 3.4.2)
- ii Markererkennung und -Tracking - im Bild rot markiert, als Ergebnis wird die Kameraposition, also die Rotation und Translation, relativ zum identifizierten AR-Marker beschrieben (siehe 3.4.1)
- iii Bestimmung der Kamera-Roboter Relation - im Bild blau markiert, ist gleichbedeutend mit der Transformation der Kameraposition aus dem zweitem Schritt in den Roboterkoordinatensystem (siehe 3.4.3)
- iv Bestimmung der Bildposition und -orientierung des virtuellen Objektes - dank der im letztem Schritt bestimmten Relation, können die Bildpositionen der Roboterzustands- und Prozessvariablen mit Hilfe von intrinsischen Kameraparametern bestimmt werden (siehe 3.2.3)

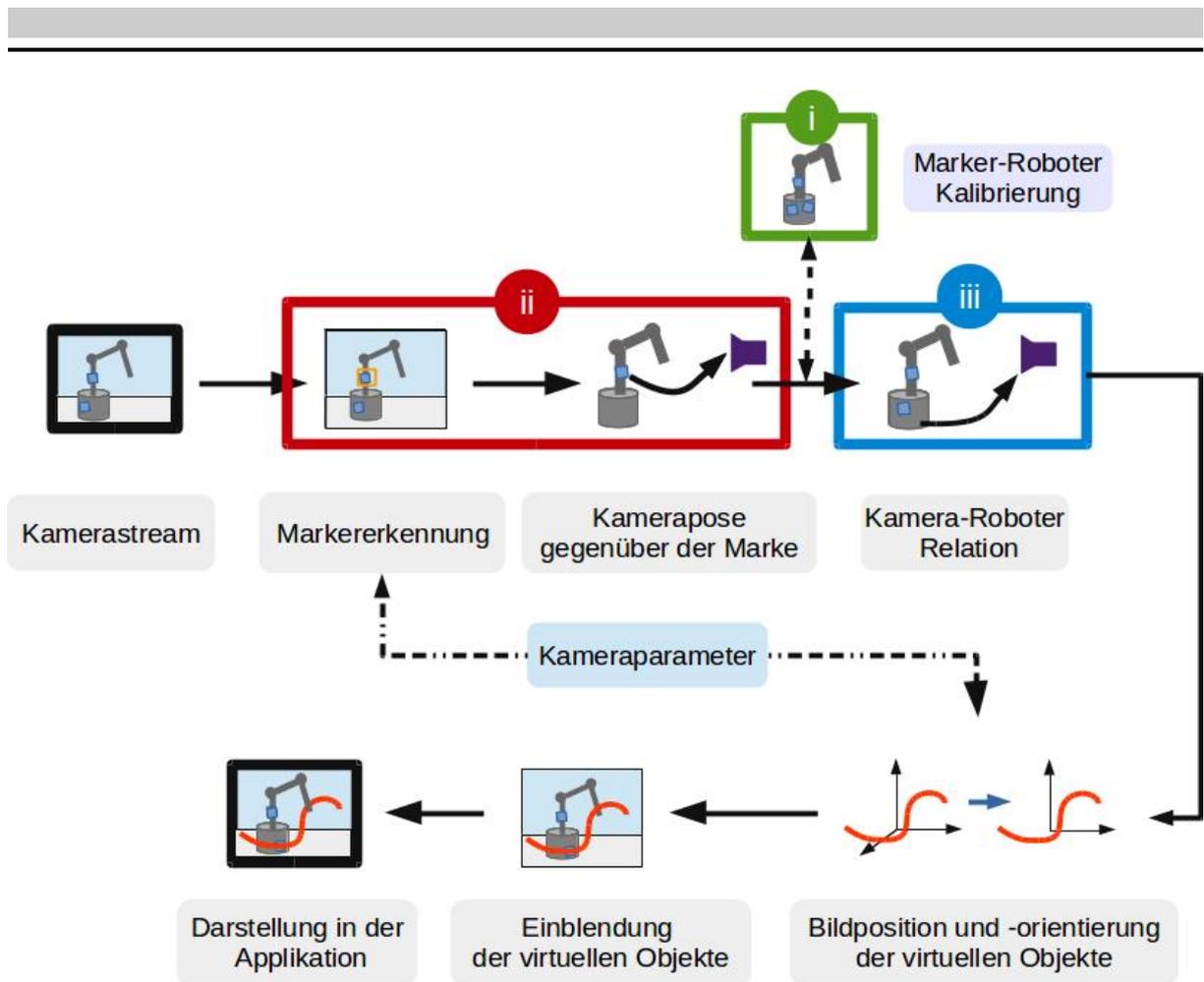


Abbildung 3.6: Zusammenhang der verschiedenen Schritte bei der Bestimmung der Kameraposition und der Objektregistrierung. Die Marker-Roboter Kalibrierung muss nur bei der Neupositionierung der Marker auf dem Roboter durchgeführt werden. Die restlichen Schritte werden für jedes Bild des Livestreams ausgeführt.

- v Einblendung der virtuellen Objekte - Durch die Projektion der 3D-Objekte auf eine 2D-Bildebene können diese in das Bild eingeblendet werden.

Anzumerken ist, dass sich während der Systemkalibrierung zumindest ein Marker im Blickfeld der Kamera befinden muss. Dabei wird der erste Schritt (in der Abbildung grün markiert) nur bei Änderung der Position der Marker durchgeführt. Die restlichen Punkte werden für jedes Bild des Videostreams ausgeführt. Zusätzlich benötigt der erste und dritte Schritt Informationen über die Anordnung der Robotergelenke.

Zu beachten ist, dass die Genauigkeit der perspektivischen Darstellung der Daten von der präzisen Angabe der Markerpositionen in einem festen Bezug zum Roboter abhängt. Bei der Bestimmung der Kameraposition im Roboterframe muss sowohl die Änderung der Kamera- als der Markerposition beachtet werden.

*Hinweise zur Benennung der Marker:* Für die Beschreibung, der in diesem Kapitel beschriebenen Konzepte werden folgende Bezeichnungen eingeführt:

- 
- i ArMarker - Ein vordefinierter Marker der auf dem Roboter positioniert wird und bei der Bestimmung der Kamera-Roboter Relation benötigt wird
  - ii CamMarker - Ein vordefinierter Marker der in der Marker-Roboter Kalibrierung zur Bestimmung der festen Kameraposition verwendet wird

---

### 3.4.1 Markerbasiertes Tracking

---

Zur Bestimmung der Kameraposition werden quadratische Marker mit bekannter Größe verwendet. Diese sind mit einem dicken, schwarzen Rahmen versehen. Das innere Muster (engl. pattern) wird auf einem weißen Hintergrund positioniert. Auf diese Art kann sicher gestellt werden, dass der Kontrast zwischen dem Marker und dem restlichem Bild, und die daraus resultierende Erkennungsrate möglichst hoch ist. Jeder Marker kann durch das innere Muster eindeutig identifiziert werden.

Um die Markerposition in einem Videostream zu verfolgen und so Rückschlüsse auf die Kameraposition ziehen zu können, müssen alle Marker zuerst in dem Videostream identifiziert werden. Zu diesem Zweck werden in jedem Kamerabild die folgenden Schritte ausgeführt [14], wobei dessen Ergebnis in der Abbildung 3.7 zu sehen ist:

- i **Transformation in binäres Bild** - Nach der Anwendung von einem Schwellwert und einem Segmentierungsalgorithmus, entsteht ein binäres (schwarz-weiß) Bild, welches die einheitlichen Bereiche des Bildes repräsentiert.
- ii **Erkennung der Geraden** - Als nächstes werden alle Liniensegmente im Bild lokalisiert. Durch eine einfache Anpassung von jeweils vier Geradenabschnitten kann entschieden werden, ob es sich bei dem jeweiligen Bildbereich um ein Viereck, einem möglichen Marker, handelt. Die Schnittpunkte definieren dabei die Position von den Ecken des Vierecks im Bild.
- iii **Pattern Normalisierung** - Ist eine viereckige Region im Bild gefunden, muss diese normalisiert, beziehungsweise entzerrt werden. Anhand der gefundenen Liniensegmente und Ecken kann eine Transformation berechnet werden, die den zusammenhängenden Bereich in eine quadratische Region umwandelt, das heißt Kanten haben die gleiche Länge und alle Winkel  $90^\circ$ . Diese Berechnung basiert auf einer Bestimmung der Homographie zwischen den gefundenen Markereckbildpunkten und bekannten Markerkoordinaten.
- iv **Marker Identifikation** - Im letzten Schritt des Markertrackings wird das innere Muster des erkannten Markers mit allen vordefinierten Vorlagen verglichen. Der Marker wird durch das Muster, für welches die größte Übereinstimmung eingetroffen war, eindeutig definiert.

Ist ein Marker im Videostream gefunden, so ist auch die Transformationsmatrix  ${}^K T_M$  zwischen der Marke und dem Kamerakoordinatensystem aus der Berechnung der Homographie bekannt.

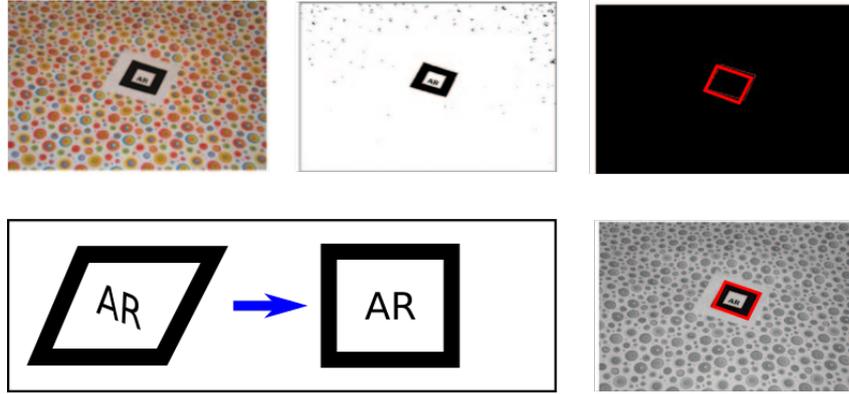


Abbildung 3.7: Markererkennung. Als erstes wird das Bild (oben links) in ein binäres transformiert (oben mitte). Im Zweitem Schritt werden Geraden identifiziert (oben rechts). Falls ein Marker gefunden wurde, wird dieser Normalisiert (unten links). In dem letztem Schritt erfolgt die Markeridentifikation (unten rechts) **Quelle:** Modifiziert nach [14]

### 3.4.2 Marker - Roboter Kalibrierung

Eine Voraussetzung zur Bestimmung der Kameraposition in Relation zum Roboter ist die Herstellung einer festen Beziehung zwischen dem Marker und Roboter. Das heißt gesucht wird die Position der  $n$  vordefinierten ArMarker auf den entsprechendem Roboterglied. Dazu wird ein semiautomatisches Verfahren entwickelt.

Es wird angenommen, dass die Kamera während der Bestimmung der Marker-Roboter Relation fest in der Umgebung des Roboters positioniert ist. Ihr Standort wird beim Start anhand eines **CamMarkers** bestimmt. Die Position des CamMarkers muss möglichst genau bzgl. eines gewählten Roboterframe (**Linkframe**) angegeben werden. Ist der gewählte Linkframe nicht mit der unbeweglichen Roboterbasis (**Basisframe**) identisch, wie in der Abbildung 3.8 (i), zu sehen ist, muss seine Position und Orientierung bzgl. des Basisframes transformiert werden:

$${}^{M_{cam}}T_B = ({}^B T_{M_{cam}})^{-1} = ({}^B T_L {}^L T_{M_{cam}})^{-1} = {}^{M_{cam}}T_L {}^L T_B,$$

wobei  $M_{cam}$  dem CamMarker,  $L$  dem Linkframe und  $B$  dem Basisframe entspricht.

Nach der Anwendung eines Markerverfolgungsalgorithmus ist die Lage des **Kameraframes** relativ zum initialen Markerframe bekannt. Sie wird beschrieben durch die Transformation  ${}^{M_{cam}}T_C$ , vergleiche mit 3.8 (ii). Die relative Kameralage zur Roboterbasis kann wie folgt bestimmt werden (siehe Abbildung 3.8 (iii)):

$${}^C T_B = ({}^{M_{init}}T_C)^{-1} {}^{M_{init}}T_B = {}^C T_{M_{init}} {}^{M_{init}}T_B = {}^C T_{M_{init}} {}^{M_{init}}T_L {}^L T_B.$$

Für die Bestimmung der Position des jeweiligen ArMarkers wird ein zusätzlicher Roboterframe **Targetframe**  $TF$ , bezüglich welcher die Marker positioniert sind, benötigt. Sobald

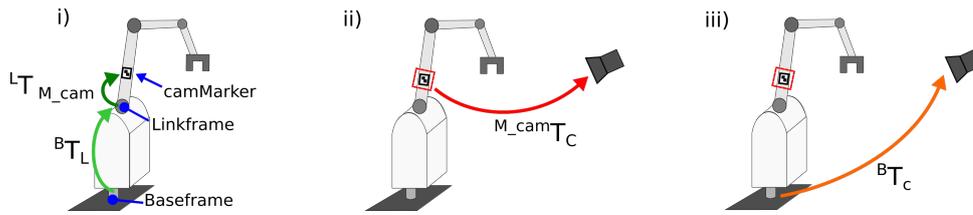


Abbildung 3.8: Schritte der Marker-Roboter Kalibrierung. Die Bestimmung dieser Relation (i) erfolgt anhand einer bekannten Transformation der Roboterbasis bzgl. dem CamMarker und (ii) der relativen Position der Kamera zu dieser. (iii) Zu bestimmende Beziehung zwischen der festen Kamera und der Roboterbasis.

ein Marker im Kamerabild erkannt wird, kann die Kameraposition relativ zu diesem Marker angegeben werden. Die Abbildung 3.9 zeigt die Anordnung der einzelnen Frames und die ausgeführten Transformationen während einer Marker-Roboter Kalibrierung. Durch die Annahme, dass die Transformation  ${}^C T_B$  zwischen der Kamera und der Roboterbasis fest ist, kann die Position des  $i$ -ten ArMarkers  $M_i$  in Beziehung zum Targetframe gebracht werden:

$${}^{M_i} T_{TF} = ({}^{M_i} T_C)^{-1} {}^{M_i} T_B = {}^C T_{M_i} {}^{M_i} T_B = {}^C T_{M_i} {}^{M_i} T_{TF} {}^{TF} T_B.$$

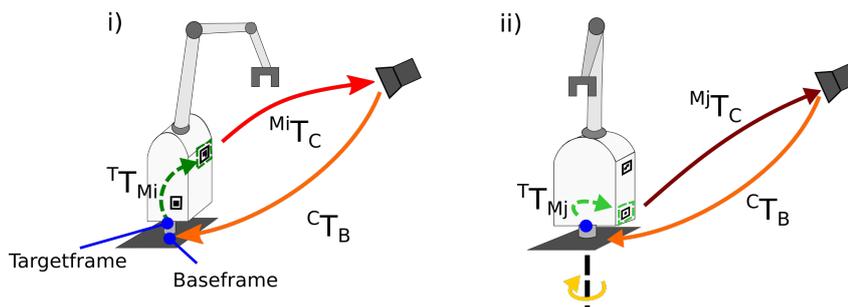


Abbildung 3.9: (i) Bestimmung der ArMarkerposition relativ zum Targetframe  ${}^{M_i} T_{TF}$ . Während der Drehung des Roboters (ii) ändert sich die Kameraposition nicht, so dass die Transformation zu allen Marker bestimmt werden kann.

Dieser Schritt sollte bei jeder Änderung der Position ArMarkers durchgeführt werden.

### 3.4.3 Bestimmung der Kamera-Roboter Relation

Dank der im letztem Schritt gefundenen Relation jedes ArMarkers  $M_i$  zur Targetframe  $TF$  ist es möglich die Position der sich bewegenden Kamera relativ zur Roboterbasis  $B$  anzugeben. Das bedeutet: Wird ein Marker  $M_i$  vom Markerverfolgungs Verfahren eindeutig identifiziert und ist die Transformationsmatrix  ${}^{M_i} T_C$  bestimmt worden, so kann die zum Marker relative Kameraposition in das Target- und demnach auch in das Basiskoordinatensystem transformiert werden. Dabei müssen folgende Fälle betrachtet werden:

- i) Die Kamera hat sich vom  $t-1$  zur  $t$  bewegt, aber die Position des Targetlinks hat sich nicht verändert (Roboter stand still, vergleiche 3.10 (i) ).
- ii) Die Kameraposition ist unverändert geblieben im Gegensatz zum Targetlink (Roboter hat sich bewegt, vergleiche 3.10 (ii)).
- iii) Die Kamera und der Targetlink haben sich im Zeitschritt zwischen  $t-1$  und  $t$  bewegt (Roboter und Kamera haben sich bewegt, vergleiche 3.10 (iii)).

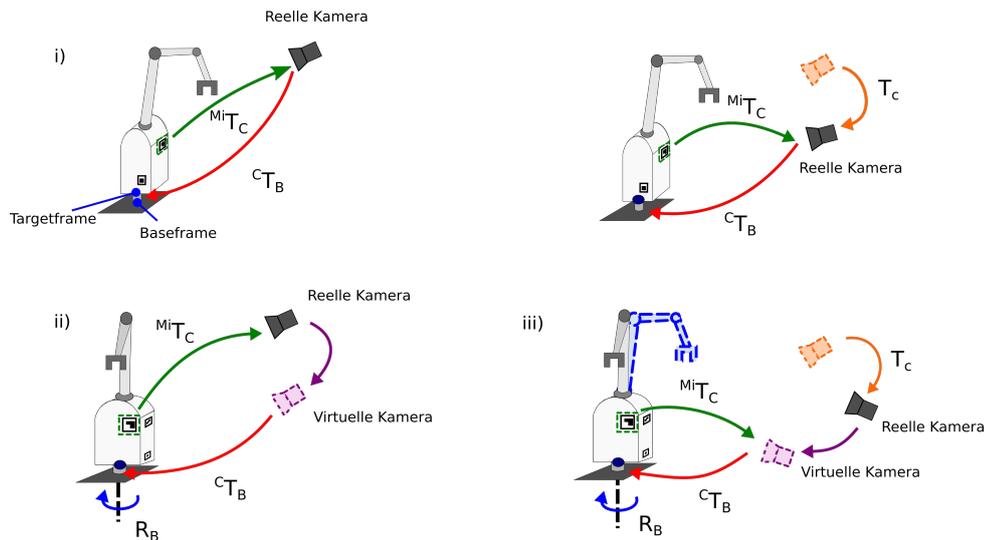


Abbildung 3.10: Bestimmung der Kameraposition bezüglich der Roboterbasis. (i) Die Transformation  $M_i T_C$  entspricht der Verschiebung der Kamera bezüglich der Marker, falls der Roboter still steht. Bei der Positionsänderung der Position der Kamera (ii) und der entsprechenden Robotergerlenken (iii) entspricht die Transformationsmatrix  $M_i T_C$  nicht der tatsächlicher Bewegung des Aufnahmeegerätes.

Im ersten Fall entspricht die Änderung der Kameraposition, der Transformation des identifizierten Markerbereichs in dem aktuellem Bild. D.h. Die Transformationsmatrix zwischen der Kamera und dem Marker entspricht der realen Verschiebung und der Rotation des Aufnahmeegerätes. Diese muss lediglich relativ zu dem Basiskoordinatensystem angegeben werden:

$${}^B T_C = {}^B T_{Tf} {}^{Tf} T_C = {}^B T_{Tf} {}^{Tf} T_{M_i} M_i T_C,$$

wobei  ${}^B T_{Tf}$  und  ${}^{Tf} T_{M_i}$  (siehe 3.4.2) im Vorfeld bekannt sein müssen.

Im zweiten und dritten Fall entspricht die Transformationsmatrix nicht der tatsächlichen Position der Kamera. Durch die Bewegung des Roboters, ändert sich die Sicht der Kamera auf den entsprechenden Marker. Da das Verfahren zur Bestimmung der Kameraposition das aktuelle Kamerabild, ohne zusätzliche Informationen bezüglich möglicher Bewegung des Roboters in der Zeit zwischen  $t-1$  und  $t$ , bearbeitet, geht es immer davon aus, dass sich das Aufnahmeegerät bewegt hat. Deswegen ist die Korrektur des Kameraframes notwendig.

Zunächst wird angenommen, dass sich die Position der Robotergelenke vom Zeitpunkt  $t-1$  bis  $t$  nicht verändert hat. Mit dieser Annahme wird die Kamera-Basis Transformation  ${}^B T_C$  wie oben bestimmt. Als nächstes muss die Rotation und Translation des Basisframes gegenüber dem Tragetframe bestimmt werden:

$$\hat{T} = {}^B T_n {}^n T_{n-1} \cdot \dots \cdot {}^0 T_{TF},$$

wobei  ${}^i T_{i-1}$  die Transformation des  $i$ -ten bzgl. des  $i-1$ -ten Gelenks des Roboters entspricht. Zum Schluss muss die Beziehung zwischen der virtuellen Position der Kamera und der Roboterbasis um diese Transformation korrigiert werden, d.h. die Kamera-Basis Transformation wird mit  $\hat{T}^{-1}$  multipliziert.

Ist die Kameratransformation gegenüber der Roboterbasis bestimmt, so können die 3D Informationen perspektivisch korrekt auf das Bild projiziert werden (siehe 3.2.3).

### 3.5 Software Architektur

Um die effiziente und zuverlässige Darstellung des Roboterzustands zu garantieren, ist eine Client-Server-Architektur ausgewählt worden. Diese erfordert mindestens zwei unabhängige Computer, welche miteinander kommunizieren und unterschiedlichen Aufgaben nachkommen [58]. Die Abbildung 4.1 zeigt die einzelnen System- und Softwarekomponenten, sowie die Kommunikation zwischen ihnen.

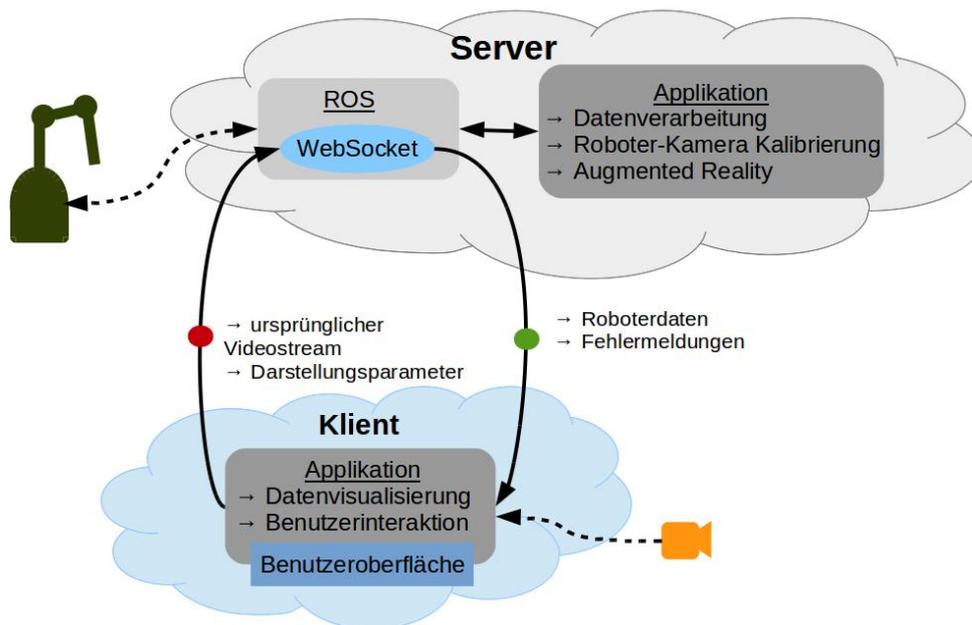


Abbildung 3.11: Systementwurf. Server-Client-Architektur

Die **Darstellungsapplikation** auf dem **Serverrechner** hat die Aufgabe die Daten, die zur Darstellung des Roboterzustands und -prozesses benötigt werden, bereitzustellen. In

---

diesem Modul wird unter anderem die Kameraposition bestimmt, welche für perspektivische Darstellung der Roboterdaten notwendig ist. Zusätzlich wird auf diesem Rechner die Kommunikationsmiddleware (*Ros*) ausgeführt, welche die Infrastruktur der Kommunikation zwischen den einzelnen Komponenten (Roboter, Klientrechner und Darstellungssaplikation) bereit stellt.

Als **Klientrechner** kann ein mobiles Endgerät (Tablet, Handy) oder ein PC verwendet werden. Die **Klientapplikation** verwaltet die Benutzerinteraktionen und stellt, die vom Serverrechner gesendeten Daten graphisch dar.

Beide Systeme kommunizieren miteinander und tauschen Daten über ein Web-Socket aus. Zum einen sendet der Klientrechner den Videostream und die Darstellungsparameter an den Server, zum anderen bekommt er die Nachrichten über den aktuellen Roboterzustand, die zur Visualisierung in der Benutzeroberfläche benötigt werden. Eine Erweiterung des Systems sieht vor, dass die Kamera an den Serverrechner angeschlossen werden kann. In diesem Fall wird auch der ursprüngliche Videostream an den Klientrechner weiter geleitet. Dies ermöglicht beispielweise eine Fernroboterinspektion mit einer fest in der Roboterumgebung positionierten Kamera.

Die **Benutzeroberfläche** bildet eine graphische Mensch-Maschine-Schnittstelle. Durch die Interaktion mit ihr kann der Anwender die Inhalte der Applikation an seine Bedürfnisse und Ansprüche anpassen. Zum Beispiel kann durch die Auswahl von graphischen Elementen der Videostream mit zusätzlichen Informationen wie die Roboterbewegungsbahn angezeigt werden oder die genaue Position der Robotergetriebe der Tabelle entnommen werden.

Die Benutzeroberfläche wird in semantisch zusammenhängende Bereiche aufgeteilt, wie die Abbildung zeigt 3.12. Das vereinfacht die Struktur und die Navigation. Es werden drei Arbeitsfelder definiert: Status-, Darstellungs- sowie Prozess/Wartungsbereich.

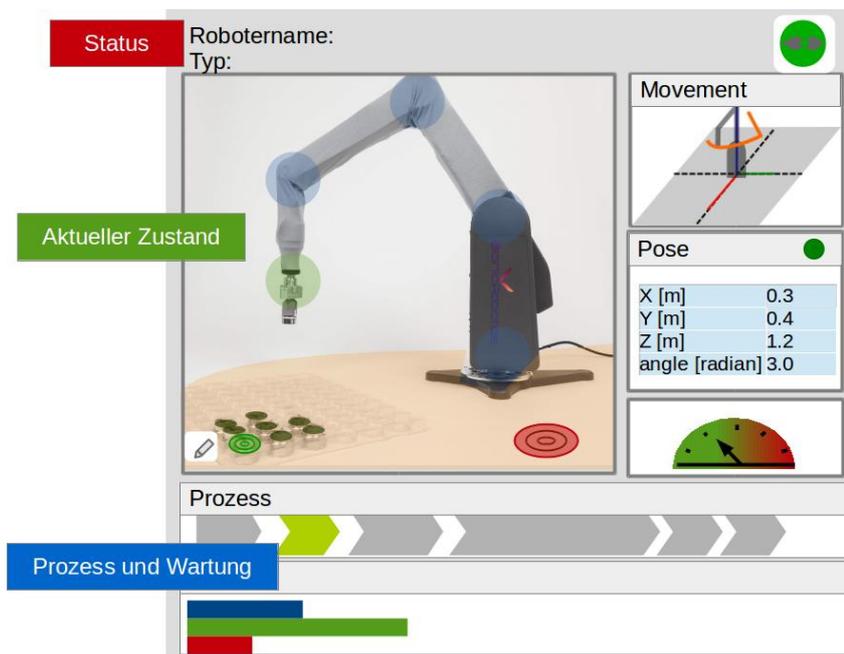


Abbildung 3.12: Entwurf der graphischen Benutzeroberfläche. Diese wird in drei semantisch zusammenhängende Bereiche aufgeteilt: Status, aktueller Zustand, Prozess und Instandhaltung, wodurch die Struktur und Navigation vereinfacht wird.

---

## 4 Umsetzung

---

In diesem Abschnitt werden die entwickelten Lösungen zur Fragestellung dieser Arbeit erläutert. Angefangen von der Systemgestaltung, über die Systemkalibrierung, bis zur entwickelten graphischen Benutzerschnittstelle.

---

### 4.1 Systemgestaltung

---

Das System zur Zustands- und Prozessvisualisierung eines Roboters besteht aus drei Hauptkomponenten: Roboter, Server und Klient. Die Roboterkomponente beinhaltet die Maschine selbst und dessen Ansteuerung eine Kontrolleinheit. Die Server- und Klientkomponenten enthalten jeweils einen Rechner. Für eine korrekte Funktionsweise des Systems ist eine Kamera notwendig. Diese kann über den Server- oder Klientrechner an das System angeschlossen werden. Die Abbildung 4.1 stellt graphisch die einzelnen Systemkomponenten und das Kommunikationsprinzip zwischen ihnen.

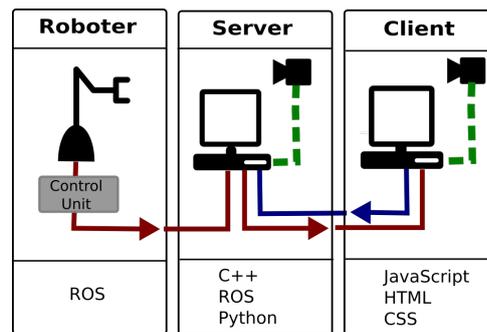


Abbildung 4.1: Systemstruktur und Kommunikationsübersicht. Das System besteht aus folgenden Komponenten: Roboter, Server und Klient. Der Roboter sendet Daten (rot markiert) an den Server und den Klientrechner, dieser tauscht Daten mit dem Server (blau markiert). Die Kamera sendet den Videostream wahlweise an den Server- oder Klientrechner. **Quelle:** Modifiziert nach [59]

Um der Anforderung an die Betriebssystemunabhängigkeit gerecht zu werden, ist die Klientsoftware als ein Web Plug-In, basierend auf einer plattformunabhängigen Technologie (JavaScript und HTML), realisiert worden. Das ermöglicht die Benutzung der Klientapplikation mit einem beliebigen Betriebssystem (z.B. Microsoft Windows, Linux oder Android). Die einzige Anforderung ist, dass das Gerät über genügend Leistung verfügt, um 3D-Elementen darstellen zu können. Dank einem Kommunikationsmodul kann diese Einheit mit dem Server kommunizieren und Daten austauschen.

Die Applikation, die auf dem Serverrechner ausgeführt wird, ist in der C++ Umgebung unter Benutzung von Ros, OpenCV und ARToolKit Bibliotheken erstellt worden. Sie beinhaltet Softwaremodule zur Kommunikation mit dem Roboter und dem Klientrechner, sowie die Bildverarbeitungs-komponente, welche die zu darstellenden Daten verarbeitet.

Die angewandte Lösung ermöglicht eine schnelle Kommunikation zwischen den verschiedenen Computerplattformen und Robotertypen.

### Datenaustausch

Die Komponenten des Systems kommunizieren miteinander mit Hilfe von *Ros*-Topics, -Services und -Parametern. Wie in der Abbildung 4.2 dargestellt wurden, werden zum einen Daten zwischen der Roboterkontrollereinheit und der Server-, sowie der Klientkomponente, zum anderen direkt zwischen dem Server- und der Klientsoftware, ausgetauscht.

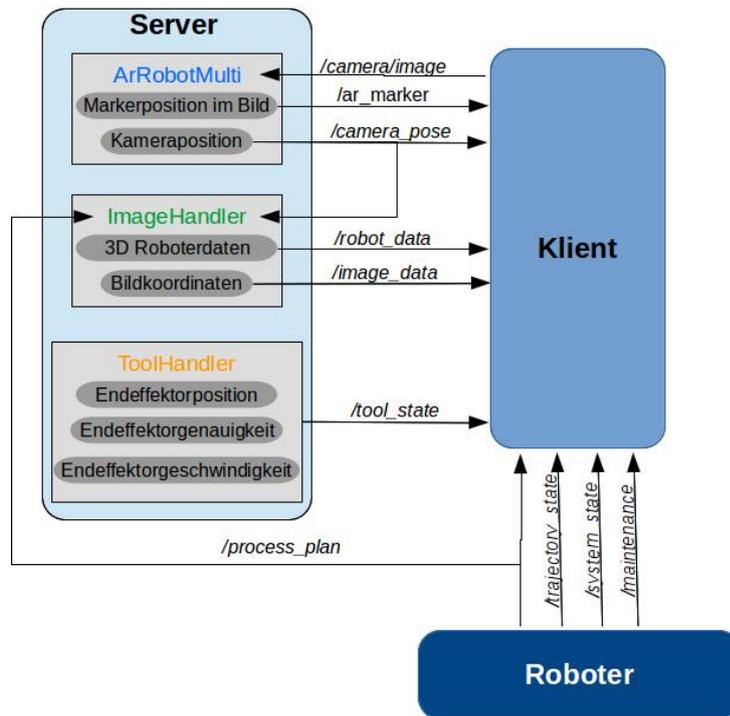


Abbildung 4.2: System Schnittstellenübersicht. Die einzelnen Komponente kommunizieren miteinander mittels Topics und Parameter (in der Abbildung nicht enthalten).

So veröffentlicht der Klient, den mit der Kamera aufgenommenen Livestream (in der Abbildung grün markiert). Dieser wird in ein komprimiertes png-Format konvertiert. Zusätzlich werden bei Benutzerinteraktionen Parameter gesetzt, welche die Berechnungen der einzelnen AR-Komponenten, wie Trajektorie oder Sicherheitsbereich, anstoßen.

Die vom Server veröffentlichten Daten werden in drei Softwarekomponenten berechnet. Dabei bildet die *ARRobotMulti* eine zentrale Rolle im entwickelten System. Es bestimmt und veröffentlicht die aktuelle Kameraposition. *ImageHandler* verwendet diese Position, um die aktuellen Roboter- und Prozessdaten zu verarbeiten und zu veröffentlichen. Dazu benötigt die Servereinheit Informationen über den aktuellen Prozessverlauf des Roboters. Diese sollen über ein Topic in einem entsprechenden Format durch die Robotersteuereinheit zur Verfügung gestellt werden.

Aus Effizienzgründen werden die Roboterdaten wie folgt gruppiert:

- 
- i Roboterdaten - sind kamerapositionunabhängig und beschreiben die räumliche Lage der Endeffektortrajektorie und des Sicherheitsbereiches des Roboters
  - ii Bilddaten - sind kamerapositionabhängig. Sie determinieren die Lage der einzelnen Darstellungselemente (Position der Gelenke, Endeffektortrajektorie) im Bild fest, d.h. 3D Koordinaten werden in ein Bildkoordinatensystem transformiert.

Die Letzte Komponente des Serversystems wird durch den *ToolHandler* gebildet. Dieser sammelt die Informationen über den Endeffektor des Roboters und publiziert sie in einer definierten Form. Im Detail sind das: die aktuelle Position, Geschwindigkeit, Kartesische- und Orientierungsgenauigkeit.

Die Schnittstelle zwischen der Roboter- und der Klienteinheit dient zum Setzen der roboterspezifischen Informationen, wie Name, Seriennummer oder Betriebsstatus. Zusätzlich hat sie die Aufgabe den Fortschritt des Roboterprozesses sowie die Instandhaltungsinformationen in einem festgelegtem Format zu übermitteln.

---

## 4.2 Systemkalibrierung und Objektregistrierung

---

Bei der Realisierung des Video-See-Through AR Systems ist die Bestimmung der Beziehungen unter Betrachtung der relativen Bewegung der einzelnen Komponenten (Kamera, Roboter, Marker) zueinander, grundlegend. Um die Anforderungen bezüglich der Genauigkeit und Effizienz des AR-Systems zu erfüllen, ist zur Markererkennung und zum Markertracking die ARToolKit Software (siehe 2.5.2) eingesetzt und modifiziert worden. Die Veränderungen, haben das Ziel die Kameraposition im Bezug auf den Roboter zu beschreiben und werden im folgendem erläutert.

Der mögliche Einsatz von mehreren Marker hat die Verwendung des Multimarkerverfahrens [60] erfordert. Dieser bestimmt die aktuelle Position der Kamera gegenüber einem Referenzpunkt, welcher durch die Angabe der relativen Transformation  $T_{i_{ref}}$  zu jedem Marker  $M_i$  festgelegt wird. Allerdings muss diese Transformation, sowie der relative Pfad zu einer Markermusterdatei, die Markergröße und der Markermittelpunkt im Vorfeld definiert und in einer Datei abgelegt werden müssen.

Aus Effizienzgründen wurden zusätzlich zu den Roboterframes, die Lage der Marker und die Kameraposition in einen *Tf*-Baum eingefügt, sodass alle bei der Bestimmung der Kamera-Roboter Relation benötigten Koordinatentransformationen mit Hilfe der Transform Bibliothek (siehe 2.5.1) bestimmt werden können.

Wie die Abbildung 4.3 zeigt, wird die Anordnung der einzelnen Roboter- und ARframes jeweils in einer Baumstruktur festgehalten. Die Aufgabe der Systemkalibrierung ist die Verbindung dieser Subbäume.

Alle Marker werden durch die Eingabe der relativen, homogenen Transformation  $T_{i_{ref}}$  (vgl. 4.3 (a)) zu dem Referenzpunkt (*ref\_frame*), bezüglich welchem die Kameraposition mittels Markertracking (siehe 3.4.1) bestimmt wird.

Um die Subbäume miteinander zu verbinden und so die Transformation zwischen dem Roboter und der der Kamera zu ermöglichen. Wird der Referenzpunkt in dem Ursprung,

des ausgewählten Gelenks sog. Tragetframe (*target\_frame*) mit der gleichen Orientierung, positioniert. So beschreibt die  $T_{i_{ref}}$  Transformation, die relative Position von dem  $i$ -ten Marker  $M_i$  bezüglich der Targetframe. Zu beachten ist, dass diese feste Transformation die Grundlage der durchzuführenden Transformation bildet und so die Genauigkeit der Überblendung der realen Objekte mit den virtuellen Inhalten definiert.

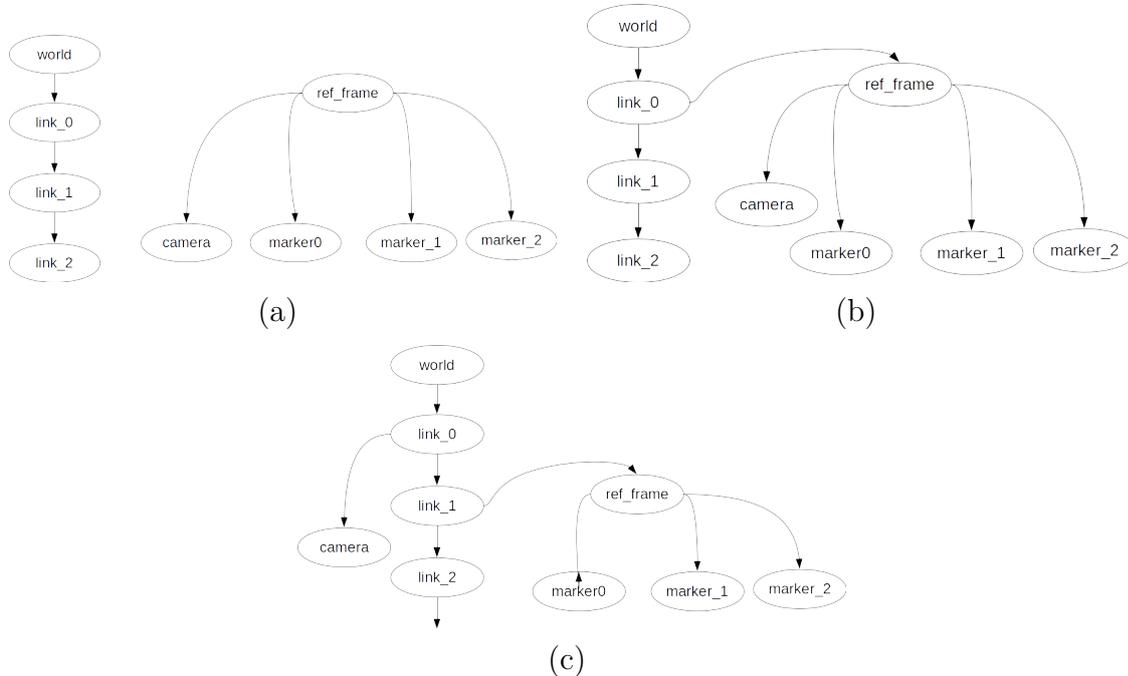


Abbildung 4.3: Die Anordnung der *Tf*-Frames, die während der Systemausführung stets automatisch bestimmt wird. (a) Es ist eine zusätzliche Referenzframe *ref\_frame* eingefügt worden, bezüglich welcher die Position der verwendeten Marker  $M_i$  durch relative Transformation  $T_{i_{ref}}$  festgelegt wird. (b) Sie bildet ein Verbindungspunkt zwischen dem Roboter und dem AR System. (c) Nach der Verbindung der Subbäume entspricht die relative Position der Marker, deren Lage auf dem Roboter. Mit Hilfe von dem dargestelltem *tf*-Baum kann diese genau bestimmt werden.

Mit der Realisierung des in Abschnitt 3.4.2 vorgestellten Verfahrens, wird der Anwender bei der Bestimmung von  $T_{i_{ref}}$  unterstützt. Er muss lediglich eine Kamera in der Roboterumgebung fest positionieren und einen vordefinierten Marker zur Bestimmung der Kameraposition in einer festen Position auf dem Roboter anbringen. Einzige Voraussetzung ist, dass die Marker  $M_i$  durch das Bewegen des Roboterarmes (z.B. Rotation der Roboterbasis) nacheinander sich in dem Sichtfeld der Kamera befinden. Wird ein Marker in dem Bild erkannt, so kann seine relative Lage bezüglich der Targetframe, mit Hilfe von dem *Tf*-Baum (vergleiche Abbildung 4.3 (c)) bestimmt werden. Dieser Schritt wird durch den Benutzer initialisiert und muss nur bei der neu Positionierung der Marker and der Roboter durchgeführt werden.

---

Sind einzelnen Subebäume in Beziehung zueinander gebracht worden, so kann die aktuelle Position der beweglichen Kamera bestimmt werden. Um die aktuelle relative Bewegung der Kamera zum Roboter zu beruicksichtigen, muss die im Multimarkerverfahren ermittelte, Kameraposition, wie im Abschnitt 3.4.3 beschrieben, transformiert werden. Kurz gesagt, es werden mit Hilfe des *Tf*-Pakets festgelegte Transformationen ausgefuhrt und so die relative Position der Kamera bezuiglich der Roboterbasis definiert.

Die Abbildung 4.3 (b) zeigt die Anordnung der *Tf*-Frames, die wuendert der Systemausfuhrung stets automatisch bestimmt wird.

### Objektregistrierung

Ist die notwendige Kalibrierung der Roboter-Kamera erfolgreich durchgefuhrt worden, so koennen die ausgewuehlten Roboterinformationen in die auf dem Monitor dargestellte Umgebung registriert werden. Folgend wird kurz die Bestimmung der Position von den Objekten im Bild anhand der Robotertrajektorie beschrieben.

Es wird vorausgesetzt, dass die Informationen ueber den Verlauf der Trajektorie des Endeffektors, die sich aus einem Roboterprozess ergeben, in Form von einem Array zur Verfuegung gestellt werden. Wobei jedes Element einem Teilprozess entspricht und die festgelegten Roboterpositionen in den Gelenkkoordinaten beinhaltet werden. Mit Hilfe von dem *urdf* (siehe 2.5.1) und der direkten Kinematik wird die kartesische Lage des Endeffektors in jedem durch die Trajektorie definiertem Punkt, sowie an den Zwischenpunkten wuendert des Prozesses bezuiglich der Roboterbasis bestimmt. Mit Hilfe des TF-Baums koennen sie in Kamerakoordinatensystem transformiert werden und anschlieuend wie im Abschnitt 3.2.3 beschrieben, die dazugehoerig Position in dem Bild gefunden werden.

---

## 4.3 Graphische Benutzerschnittstelle

---

Die Struktur der Benutzeroberflaeche ist absichtlich einfach gehalten. Die Struktur der Benutzeroberflaeche ermoeglicht dem Anwender jederzeit den Ueberblick ueber die ganze Funktionalitaet der Anwendung im die verfuegbaren Roboterdaten im Auge zu behalten. Durch das Auswuehlen der einzelnen Bedienelemente kann er selber entscheiden welche Informationen in jeweiligen Bereichen sichtbar sind.

Die graphische Oberflaeche ist in vier thematisch zusammenhaengende Bereiche aufgeteilt (Abb. 4.3):

- i Status - In diesem Bereich werden allgemeine Informationen ueber den Roboter, wie Modellname und Seriennummer dargestellt. Zusuetzlich wird der Benutzer, mit Hilfe von Symbolen, ueber den Betriebszustand des Roboters informiert. Die Verbindung mit dem Servercomputer kann durch das Druecken des rechten Buttons aufgebaut oder unterbrochen werden.
- ii Aktueller Roboterzustand - Auf der rechten Seite werden die aktuellen Roboterdaten visualisiert. Durch klicken im entsprechenden Bereich kann der Benutzer selbst entscheiden welche Daten er angezeigt bekommt. Der Anwender kann die 3D Darstellung der aktuellen Roboterposition und Trajektorie wuehlen. Durch das Fokussieren



Abbildung 4.4: Umsetzung der graphischen Benutzeroberfläche, welche in vier semantisch zusammenhängende Bereiche aufgeteilt ist: Status, Aktueller Roboterzustand, Prozess und Instandhaltung, sowie Logging

des Tabs „Pose“ werden Informationen über die exakte Position, sowie die momentane Genauigkeit des Endeffektors sichtbar. Schließlich kann der Benutzer die aktuelle Geschwindigkeit des Roboters durch das auswählen des „Geschwindigkeit“ Reiters einsehen.

Hauptelement dieses Bereiches bildet jedoch der Videostream, der mit zusätzlichen Informationen, wie z.B. der Trajektorie oder der Ablageposition ergänzt werden kann. Der Anwender kann durch Auswahl der entsprechenden Elemente entscheiden mit welchen Daten das Echtzeit Video überlagert wird.

- iii Roboterprozess und Wartung - Der letzte Bereich ist in Prozess und Wartung des Roboters aufgeteilt. Der Benutzer kann dort die Informationen über den aktuellen Zustand des Prozesses, z.B. die geplante Dauer oder Name des aktuell ausgeführten Elements einsehen.

---

Der Bereich Wartung bildet den Verlauf des Roboterzustands, über seine ganze Betriebszeit ab. Es werden Daten wie Gelenkgenauigkeit, oder mittlere Ausführungsdauer des Prozesses während der Arbeitsvorgänge benutzerfreundlich visualisiert.

iv Logging - Der Benutzer wird in einer Konsole über Erfolge und Misserfolge von Aktionen, z.B. Verbindung mit dem Serverrechner, informiert. Diese kann durch Auswahl des permanent sichtbaren Buttons ein- und ausgeblendet werden.

Die genaue Funktionalität und die graphische Umsetzung des Visualisierung von dem Roboterzustand und -prozess wird in dem Abschnitt 5.1 beschrieben.



---

## 5 Evaluation

---

In diesem Kapitel wird die Tauglichkeit des vorgestellten Ansatzes am konkreten Beispiel eines für die sichere Mensch-Roboter Kollaboration entwickelten Leichtbauroboter evaluiert. Zusätzlich wird der Einfluss welcher in einer entwickelten Benutzeroberfläche dargestellten Informationen auf den Anwender anhand von einer Benutzerstudie bewertet.

---

### 5.1 Darstellung des Zustands und Prozesses von dem BioRobArm

---

Der **BioRob-Arm** [3] ist ein gelenkelastischer, über die Seile angetriebener Leichtbauroboter der Firma BionicRobotics mit fünf Drehgelenken (vergleiche Abbildung 5.1). Die Konstruktion ist an menschlichen Muskel-Skelett-Apparat orientiert, was sich nicht nur in dem Antriebskonzept (Seile und Feder), sondern auch in den Ausmaßen und der Traglast bemerkbar macht. Dank seiner Bauweise können Objekte bis zu 800 Gramm, bei einem Robotergewicht von nur acht Kilogramm, manipuliert werden.

Der Antrieb des BioRob befindet sich nahe der Basis, wobei die Drehmomente zur Aktuierung der Gelenke über Seile übertragen werden. Dies ermöglicht die Reduktion des Gewichts des sich bewegenden Arms. Dessen Folge ist die Reduktion der Kontaktkräfte im Falle einer Kollision. Das bedeutet der Roboter kann ohne weitere Schutzmaßnahmen im direkten Umfeld des Menschen eingesetzt werden.

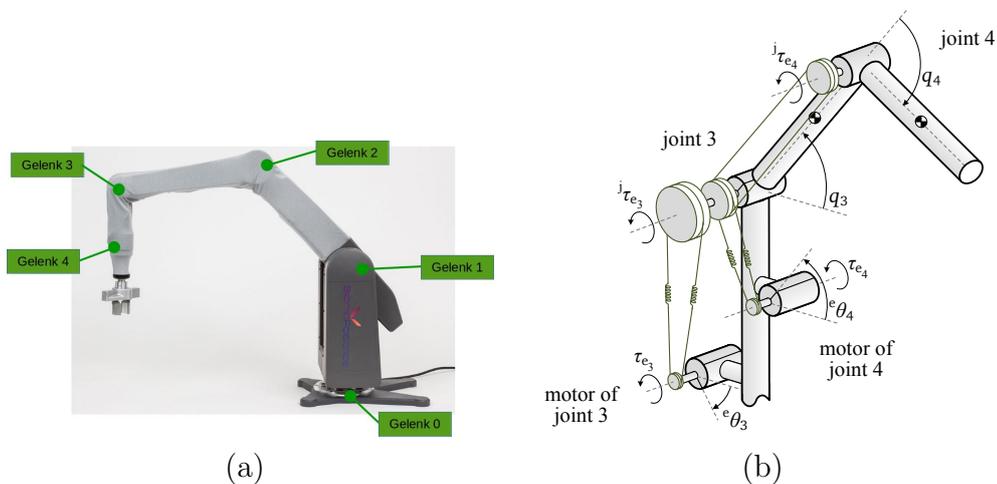


Abbildung 5.1: Mechanische Struktur des BioRob Arms. a) Der Roboter hat 5 Drehgelenke. b) Die Gelenke eins und zwei, sowie zwei und drei sind jeweils mit einem Antriebsseil verbunden. Die Antriebsmotoren befinden sich entsprechend hinter der Rotationsachse des ersten Gelenks, oder in der Roboterbasis. **Quelle:** [3]

Durch die einfache Montage, Einrichtung (geführtes Programmieren von Bewegungen) und das geringe Eigengewicht eignet sich der BioRob besonders für „Pick-and-Place“

Aufgaben, wie beispielsweise Be- und Entladen von Maschinen, oder Einsortieren von Produkten in Paletten. Dank einer intuitiven Programmierung kann ein neuer Prozess in wenigen Minuten definiert und ausgeführt werden.

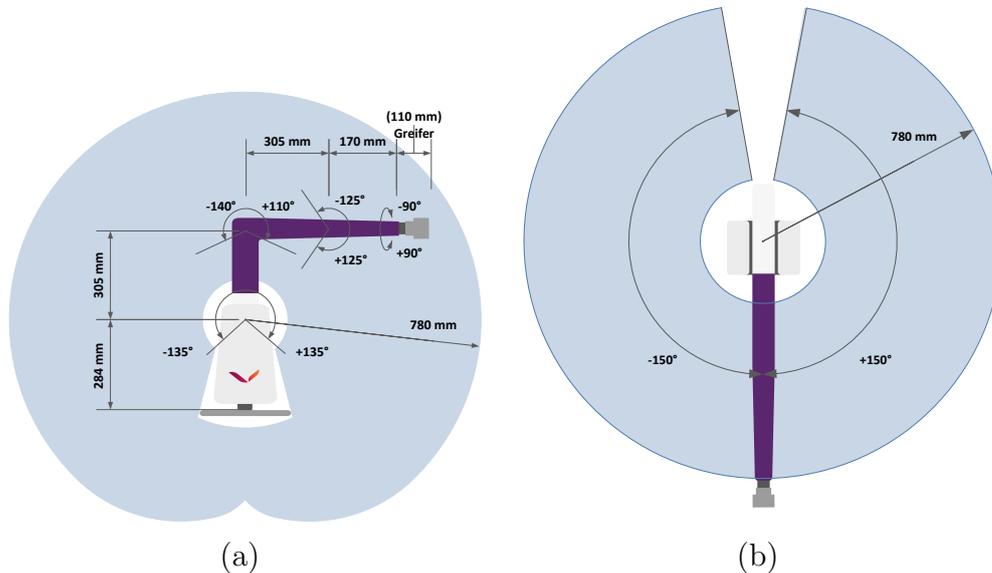


Abbildung 5.2: Bewegungsbereich des BioRob-Arms. (a) Seitenansicht und (b) Draufsicht  
**Quelle:** [3]

Der BioRob-Arm verfügt über folgende Schnittstellen: intuitive GUI, C \ C++ API, Ros, digitale Ein- und Ausgänge, MQTT, Seriell. In Verbindung mit dem *urdf* Modell werden Informationen über die aktuelle Position der einzelnen Roboterframes im *Tf-Baum* des BioRobs aktualisiert.

---

### 5.1.1 Experiment

---

Mit einem einfachen Experiment soll die Darstellung des Roboterzustands und des Roboterprozesses in der entwickelten AR-Unterstützten, graphischen Benutzeroberfläche evaluiert werden. Der Roboter *BiorobUltraV2X5\_HG* führt dabei einen Prozess aus, welcher sich aus drei verschiedenen Trajektorien zusammensetzt. Während dieser Bewegung nimmt er ein Objekt an der Startposition auf und führt eine kreisförmige Bewegung zu einer Zwischenstation, an der das Element geprüft wird, durch. Anschließend fährt der Arm zur Zielposition und legt das Element ab. Die geplante Dauer des Prozesses beträgt dabei ca. 36.8. Sekunden. Die Abbildung 5.4 zeigt die Umgebung des Roboters und die Position der einzelnen Stationen, sowie den Marker während der Ausführung des Experiments.

---

### 5.1.2 Systemkalibrierung

---

Zur Markerekenung und Bestimmung der Kameraposition wird eine handelsübliche *C270* Webkamera der Firma Logitech mit einer Auflösung von  $1280 \times 720$  Pixel verwendet.

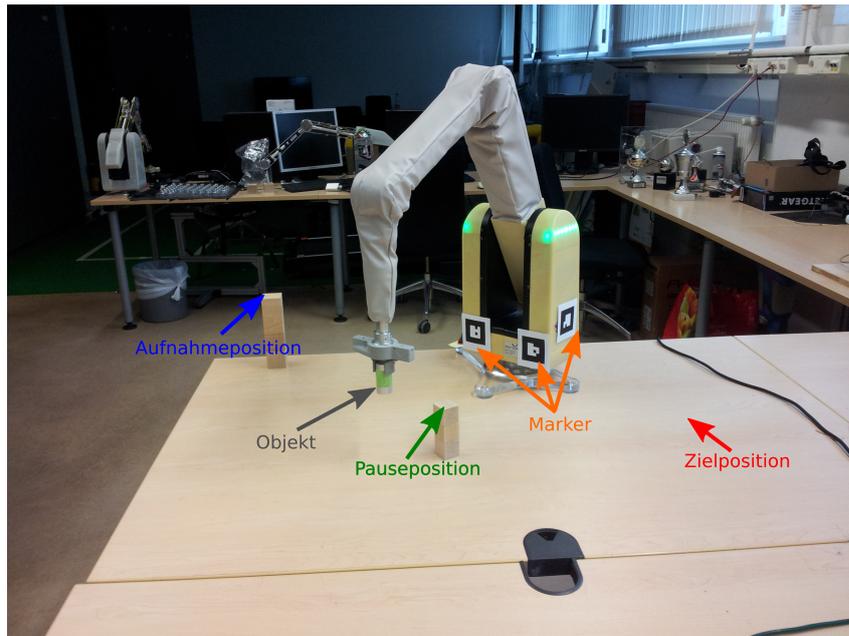


Abbildung 5.3: Der Biorob-Arm und seine Arbeitsumgebung. Der Roboter soll ein Objekt von der Startposition aufnehmen und zur Zielposition transportieren. Auf dem Robotertorso sind 6 Marker positioniert worden: zwei auf der Vorder-, zwei auf der Rückseite und jeweils ein Marker auf der rechten und linken Seite .

Auf dem sich drehenden ersten quaderähnlichen Roboterglied sind sechs Marker mit einer Seitengröße von 5 cm positioniert. Die jeweiligen Marker können eindeutig durch den Vergleich des inneren Musters identifiziert werden. Dessen Position auf dem Roboter ist mit Hilfe von dem in Abschnitt 3.4.2 vorgestellten Verfahren bestimmt worden. Die Abbildung 5.4 zeigt die Einzelnen Schritte des Verfahrens welche vor der Ausführung des Experiments zur Marker-Roboter Kalibrierung ausgeführt wurden.

---

### 5.1.3 Roboterintention und -zustand

---

Die Abbildung 5.5 zeigt die Standardansicht der Benutzeroberfläche, falls:

- i die Klientapplikation - mit dem Serverrechner verbunden ist
- ii der Roboter - ein Prozess ausführt

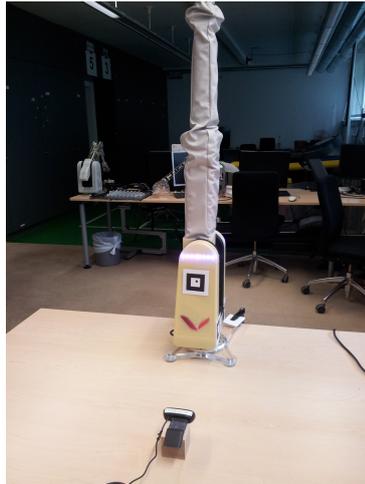
Die Hauptfunktion dieser Benutzeroberfläche ist die benutzerfreundliche Darstellung des Roboterzustands und -prozesses. Als eine zusätzliche Funktionalität wird die Unterstützung des Benutzers während der Roboter-Marker Kalibrierung gezählt.

Die Benutzeroberfläche setzt sich zusammen aus den folgenden Bereichen: Status, aktueller Zustand, Prozess, Wartung, sowie Logging.

Standardmäßig wird beim Öffnen der Applikation eine Verbindung mit einem Serverrechner über ein lokalen Port aufgebaut. Die URL-Serveradresse kann jederzeit durch den



(a)



(b)



(c)

Abbildung 5.4: Marker-Roboter Kalibrierung am Beispiel des Biorob-Arms. (a) Die Kameraposition wird in Bezug auf den CamMarker, dessen Position bezüglich der Biorobbasis bekannt war, bestimmt und im weiteren Verlauf des Verfahrens als fest angenommen, (b)(c) Als nächstes wurde der Roboterarm so bewegt, dass nacheinander alle sechs, auf dem Roboter positionierten ARMarker zu sehen waren. Konnte ein Marker eindeutig identifiziert werden, so wurde seine Position bzgl. des ersten Robotergelenks automatisch bestimmt und gespeichert.

Anwender geändert werden. Ist die Servereinheit nicht verfügbar, so wird diese Information in Form eines entsprechenden Symbols an den Benutzer übermittelt. Zusätzlich werden alle Elemente der Oberfläche (bis auf den Statusbereich) ausgeblendet.

Besteht eine Verbindung zu dem Roboter, so ist in dem oberen Bereich des Statusbereichs der Serienname *BioRobUltraV2X5\_HG*, die Seriennummer, der kinematischer Typ, sowie der Betriebsstatus eingeblendet.

Die Übertragung der mit der Kamera live aufgenommenen Bilder bildet das zentrale Element der Anwendung. Dieses kann mit den roboter- und prozessspezifischen Informationen überblendet werden. Der Benutzer hat die Möglichkeit die angezeigten Informationen auszuwählen. Die einzige Voraussetzung für das präzise Einblenden der Informationen ist, dass mindestens ein Marker im aktuellen Bild identifiziert wurde. In diesem Fall, wird dessen Position gelb hervorgehoben. Wählt der Benutzer ein, oder mehrere Elemente, die in dem Videostream angezeigt werden, so aktiviert sich die Transparenzfunktion dieser Elemente (bis zu 70%). Dabei hängt die Bildposition dieser Element von der Position des Aufnahmegerätes ab. Die Abbildung 5.6 zeigt eine Trajektorie, die dem aktuellen Roboterprozess entspricht. Diese kann aus verschiedenen Perspektiven, die sich durch die Verschiebung der Kameraposition ergaben, betrachtet werden. Neben der Trajektorie stehen dem Anwender folgende Informationen zur Verfügung:

- i Joints - Position der einzelnen Robotergelenke

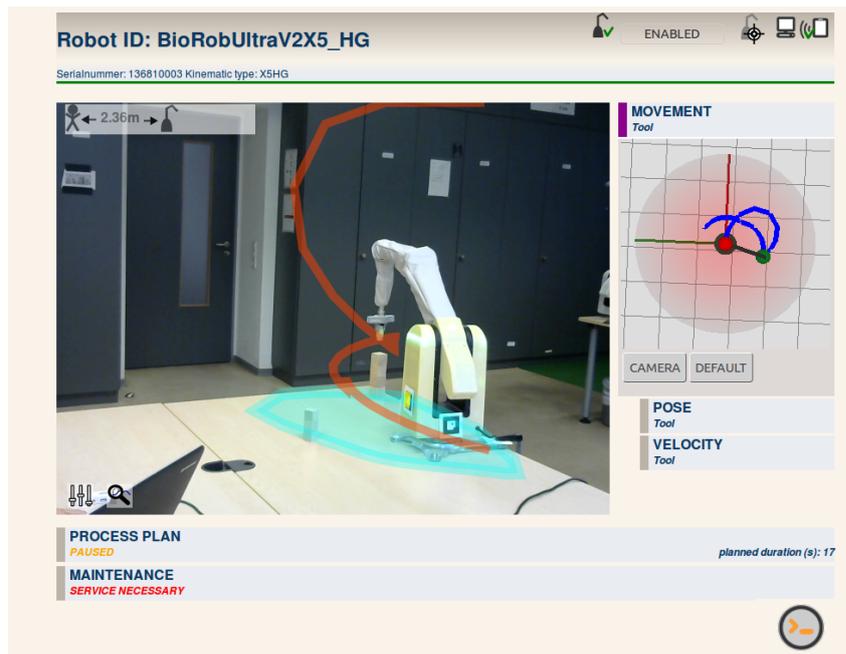
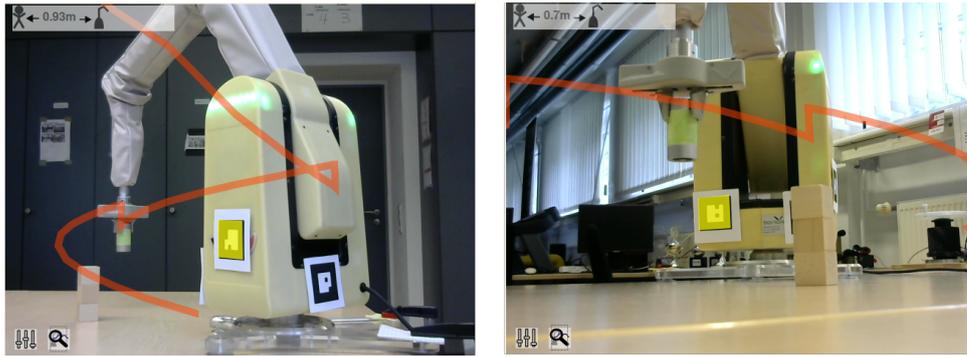


Abbildung 5.5: Sicht auf die Benutzeroberfläche. Nach der Initialisierung des Roboters, wird Name, Serinennummer, sowie Betriebsstatus in der Applikation dargestellt.

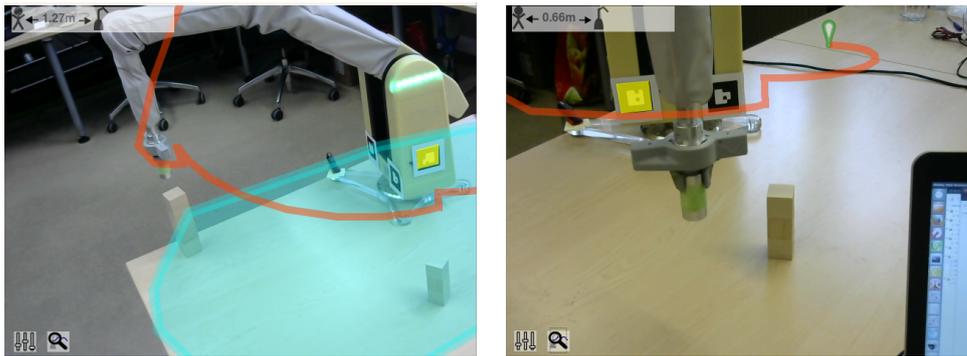
- ii Trajectory - die Endeffektortrajektorie
- iii Parttrajectory - der aktuelle Abschnitt der Trajektorie des Endeffektors
- iv Safespace -  $2D$  Projektion, des Bereiches zwischen dem Endeffektor und dem Roboter während der Ausführung des gesamten Prozesses
- v Startplace - Aufnahmeposition des Endeffektors
- vi Goalplace - Ablageposition des Endeffektors

Verschiedene Blickwinkel ermöglichen die Erkennung potentieller Gefahren (z.B. Kollisionen) oder Überprüfung der korrekten Ausführung des Prozesses. Falls der Roboter mit einem Objekt kollidiert, wird dieses Ereignis dem Benutzer im Videostream in Form eines roten Textes mitgeteilt. Zusätzlich wird in der rechten oberen Ecke des aktuellen Bildes permanent der kleinste Abstand zwischen dem Menschen und dem Roboter dargestellt.

Die rechte Seite beinhaltet eine Reihe von Tabs, welche eine andere Sicht auf den aktuellen Roboterstatus und seine Intention bieten. Durch die Auswahl der „Movement“ Schaltfläche kann der Benutzer den Roboter die aktuelle, sowie die nächste definierte Position und die Endeffektortrajektorie in einer  $3D$  Umgebung betrachten. Durch das Drücken der entsprechenden Maustaste und ihre Bewegung bietet sich dem Nutzer die Möglichkeit den aktuellen Modellausschnitt zu Verschieben, zu Rotieren oder zu Skalieren. Dank einer „Drag-and-Drop“ Funktion kann die Ansicht in die Mitte des Webbrowsers (Plug-Ins) verschoben werden. Zusätzliche Hilfe bieten Buttons zur Einstellung des Blickwinkels auf den aktuellen Bereich. Zum einen ist das die Sicht auf die positive X-Achse



(a)



(b)

Abbildung 5.6: Visualisierung des BioRob-Arms. Intention mit dem Plug-In. Aktuelle Bewegungsbahn des Endeffektors (a) aus verschiedenen Perspektiven bzw. mit einem einblendeten (b) Sicherheitsbereich, oder der Zielposition.

des Roboters, zum anderen die Perspektive, die der realen Kamera entspricht. Damit wird der Benutzer beim Erforschen der verschiedenen Einstellungen unterstützt.

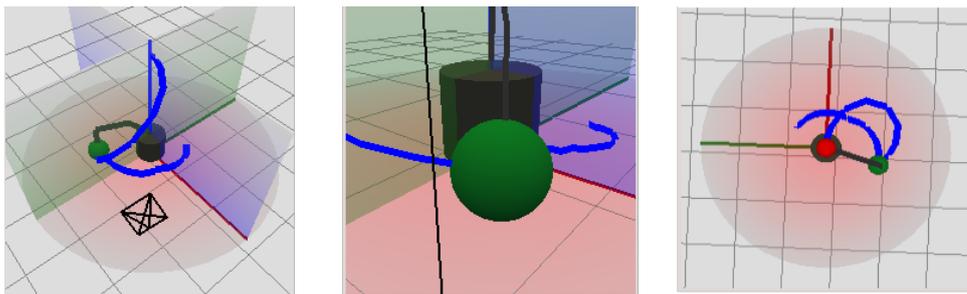


Abbildung 5.7: Movementschaltfläche. 3D Ansicht auf die aktuelle Roboter- und Kameraposition, sowie die Bewegungsbahn.

Die exakte und die nächste definierte Position des Endeffektors relativ zu der Roboterbasis, sowie die momentane Genauigkeit der Bewegung können in dem Abschnitt „Pose“ entnommen werden. Befindet sich der Roboter in der Ausführung eines Prozesses, so wird

die momentane Geschwindigkeit vom Tool benutzerfreundlich mit Hilfe eines Tachometers (Abschnitt „Velocity“) dargestellt.

Durch die verschiedenen Darstellungsformen kann der Benutzer die Ansicht an seine eigenen Bedürfnissen und Erfahrungen anpassen. Er kann jeder Zeit die angezeigte Menge an Informationen selbst steuern. Dadurch kann der Anwender seine eigene Aufmerksamkeit auf die für ihn wichtigen Elemente beschränken.

Im Bereich „Process“ sieht der Benutzer den Ausführungszustand (aktuell *paused*), die Wiederholungsinformation (zur Zeit wird der Prozess nicht wiederholt) und die geplante Dauer von 36.8 Sekunden. Alle Prozesselemente sind jeweils als ein Rechteck, mit einer Seitenlänge proportional zur jeweiligen Dauer, dargestellt. Dieser beinhaltet den Namen des Prozesselements und seine geplante Ausführungsdauer. Über ein Piktogramm wird der Typ des Elements (z.B. Trajektorie) visualisiert. Der aktuell ausgeführte Teilprozess wird grün markiert (in der Abbildung 5.8: *Prüfen*). Durch das Anklicken eines Rechtecks, werden zusätzliche Informationen eingeblendet.

Diese vereinfachte Darstellung, ist intuitiv und auch für unerfahrene Benutzer verständlich. Der Anwender bekommt eine Übersicht über die Länge des gesamten Prozesses, sowie Dauer der einzelnen Abschnitte. Des weiteren kann er jeder Zeit den Status des Prozesses erkennen, wodurch keine Unsicherheiten bei den Bewegungspausen des Roboters entstehen.



Abbildung 5.8: Prozessplan des Biorobs.

Die für die Überprüfung des Wartungszustands des Roboters wichtigen Informationen werden durch die Auswahl der „Maintenance“ Schaltfläche aktiviert. In diesem Bereich werden alle zur Verfügung stehenden Informationen über den Roboter, die langfristig aufgezeichnet werden, benutzerfreundlich dargestellt. Im Fall des Biorob-Arms (siehe Abb. 5.9) sind das:

- i Betriebszeit
- ii Misserfolge - Kommunikationsausfälle, Kollisionen, Softwarefehler
- iii Genauigkeiten - Kartesische-, Orientations-, Joint
- iv Prozessdaten - Prozessdauer, Prozessfehler
- v Seildehnung

Die Verschlechterung der Genauigkeiten ermöglicht Rückschlüsse über die maximal zu erreichende Genauigkeit während der Ausführung eines neuen Prozesses. Mit steigender



Abbildung 5.9: Visualisierung des BioRob-Arms Zustands. Die Wartungsinformationen werden in Form von Balken und Liniendiagrammen dargestellt.

Betriebszeit und Prozessdauer wächst die Abnutzung des Roboters. Sind die Seile nicht genügend gespannt, können bei der Ausführung der Bewegung Ungenauigkeiten auftreten. Durch die Darstellung der einzelnen Größen in Form von (Linien- und Balken-) Diagrammen, kann der Benutzer die eventuelle Verschlechterung des Roboterzustands früh erkennen und entsprechende Maßnahmen (beispielsweise Anpassungen am Prozess) durchführen. Zusätzlich wird dem Benutzer die Notwendigkeit der Durchführung von Wartungsarbeiten mitgeteilt. Um die Aufmerksamkeit des Benutzers auf diese Tatsache zu lenken wird diese Information rot hervorgehoben und ist auch dann sichtbar, wenn die Wartungsinformationen ausgeblendet sind.

## 5.2 Einfluss der dargestellten Informationen - Benutzerstudie

In der Studie wurde der in dieser Arbeit vorgestellte Ansatz zur Visualisierung des Roboterzustands und -prozesses evaluiert. Zu diesem Zweck wurden folgende Fragestellungen untersucht:

- 
- i Üben die angewendeten Anzeigen Einfluss auf das Situationsbewusstsein des Benutzers aus?
  - ii Welchen Einfluss haben die dargestellten Informationen auf die Wahrnehmung des Roboterzustandes?
  - iii Welche Rolle spielen die verschiedenen Informationen auf das Wohl- und Sicherheitsempfinden des Anwenders?
- 

### 5.2.1 Methoden

---

Zur Beantwortung der Fragestellungen wurde ein Laborexperiment durchgeführt. Es bildet die Grundlage der Evaluation. An der Studie nahmen 19 Probanden teil.

Vor der Durchführung des Experiments wurde jeder Proband gebeten, Fragen zu den Themen Robotik, Augmented Reality und Technik zu beantworten. Dieses diente der Ermittlung des Wissensstandes vom Experimentteilnehmer auf diesen Gebieten. Das Situationsbewusstsein ist durch das „Einfrieren“ des Experiments untersucht worden. Das bedeutet die Ausführung der Aufgabe wurde pausiert.

Das Szenario des Experiments ist an kooperativen Industriearbeitsplätze, an denen der Roboter in der nächsten Nähe von oder mit dem Menschen zusammenarbeitet, angelehnt. Während der Durchführung der Aufgabe wurde der Benutzer gebeten eine Form aus den Legeteilen des Spiels Tangram mit Hilfe des Roboters nachzubilden. Das Spiel besteht aus sieben unterschiedlich farbigen geometrischen Formen (Quadrat, Parallelogramm und fünf Dreiecke), die in verschiedenen Gestalten gelegt werden können. Die Elemente werden dabei durch eine einfache Aufreihung nebeneinander zu einem Muster zusammengefügt.

Zur Bewältigung dieser Aufgabe wurde vor dem Probanden eine Mustervorlage platziert. Der Roboter hat die Spielelemente nacheinander in einer vordefinierten Reihenfolge gegriffen und in der Nähe der Testperson abgelegt. Die Testperson sollte das Teil anschließend nehmen und an der richtigen Stelle des Musters platzieren.

Die Ausführung der Aufgabe ist zweimal zu bestimmten Zeitpunkten, die dem Probanden nicht bekannt waren, gestoppt worden. Während dieser Unterbrechungen ist jede Person gebeten worden, Fragen zu der sich unmittelbar zuvor und danach ergebenden Situation zu beantworten. Die Lösung der Aufgabe war umgehend nach der Beantwortung aller Fragen fortgesetzt worden. In dem Zeitraum zwischen dem Anfang des Experiments und der ersten Unterbrechung, hatte der Proband, neben dem Roboter selbst, keine weiteren Informationsquellen. Erst nach der ersten Pause stand die während dieser Arbeit entwickelte Applikation, die den Informationsfluss von Roboter zu Menschen ergänzt, zusammen mit der Benutzeroberfläche, zur Verfügung.

Nach Abschluss der Roboterexperimente wurden die Testperson gebeten Fragen durch Ankreuzen von Skalenwerten zur Selbsteinschätzung, sowie der Akzeptanz der Applikation zu beantworten. Mit diesem Vorgehen sind subjektive, sowie objektive Kenntnisse der Testperson erfasst worden.

---

## 5.2.2 Versuchsaufbau

---

Der Versuch fand jeweils in einem Labor der Firma Bionic Robotics und des Fachgebiets SIM der Technischen Universität Darmstadt statt. Auf dem Arbeitstisch wurde ein BioRobUltraV2X5\_HG der Firma Bionic-Robotics und ein Tray mit den Spielelementen platziert (vergleiche Abbildung 5.10). Um das Greifen der Teile mit dem Roboterarm zu ermöglichen wurde auf jedem Spielelement ein Holzbauklotz angebracht. Zusätzlich wurde auf der Arbeitsfläche zwischen Roboter und Kamera ein Marker positioniert, die zur Bestimmung der Kameraposition diente. In der weiteren Umgebung wurden keine anderen Markierungen angebracht. Die verschiedenen auf dem Tisch platzierten Objekte, füllten die Szene innerhalb des für den Prozess relevanten Arbeitsbereich des Roboters, so dass Fragen zu der eventuellen Kollision gestellt werden konnten.

Die Versuchsperson wurde angewiesen, an einem Stuhl auf dem anderen Ende des Arbeitstisches Platz zu nehmen, sodass die Teilnehmer und der Roboter sich den Arbeitsbereich teilten.

Nach der ersten Unterbrechung der Prozess wurde ein Laptop mit der entwickelten Applikation und eine Kamera auf dem Arbeitstisch positioniert, so dass die Probanden zur restlichen Lösung der Aufgabe das Verhalten des Roboters über diese verfolgen konnten.

Die gesamte Bewegung des Roboters dauerte ca. 4 Minuten. Der Roboter transportierte die Elemente in der folgenden Reihenfolge: gelbes, rotes, violettes, weißes, grünes, silbernes und blaues Element. Wobei, die jeweilige Ablage des Spielelements und so die auszuführende Endeffektortrajektorie, sich voneinander unterscheideten. Dadurch war sicher gestellt, dass kein Lerneffekt auftritt.

In der Anwendung wurde jeweils nur ein Abschnitt der Endeffektortrajektorie aus dem Gesamtprozess angezeigt. Nähmlich jeweils nur der Verlauf der Bewegung zur Auf- oder Ablage des aktuellen Elements.

---

## 5.2.3 Versuchsablauf

---

Nach Erläuterung der Probanden über die Studienmotivation und deren Ziel, wurde die Testperson in Rahmen einer Vorstudie gebeten, Fragen zu Nutzung des Roboters, Augmented Reality und Computernutzung zu beantworten.

Im Anschluss wurde den Testpersonen die Applikation und deren Funktionalität kurz vorgestellt. Danach konnte jede Person innerhalb von bis zu 5 Minuten die Applikation eigenständig erforschen und fragen zu ihr stellen.

Daraufhin folgte die erste schriftliche Anweisung, in der die konkrete Aufgabe erklärt wurde. Nachdem sichergestellt worden ist, dass die Person die Instruktionen verstanden hat, ist der Roboterprozess gestartet worden.

In einem definiertem Zeitpunkt wurde der Proband gebeten, die Durchführung seine Aufgabe zu unterbrechen. Gleichzeitig wurde der Roboter pausiert worden. Während dieser Unterbrechung hat der Teilnehmer die Fragen zu der aktuellen und zukünftigen

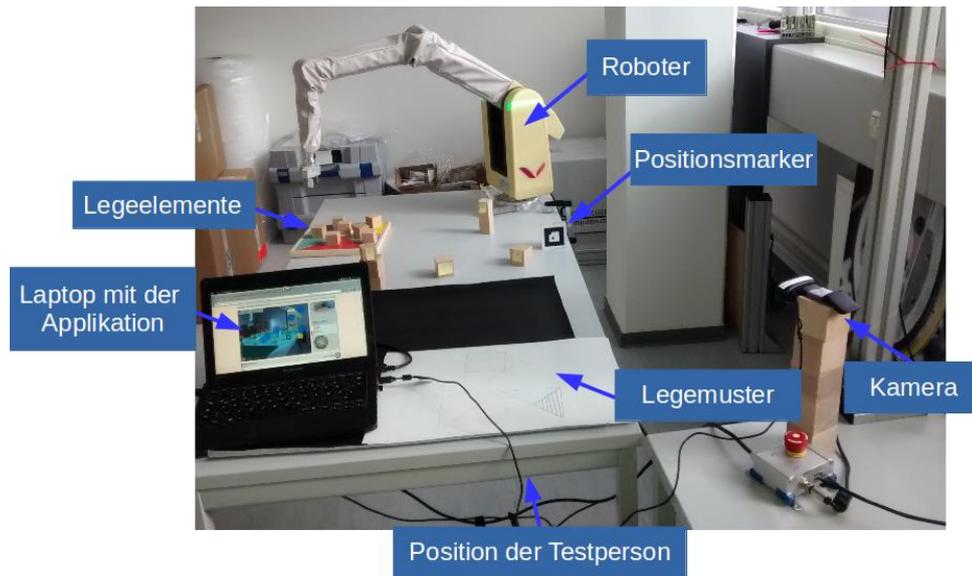


Abbildung 5.10: Versuchsaufbau. Der Arbeitstisch mit einem Roboter und den Spielelementen in dem Labor der Firma Bionic Robotics.

Situation beantwortet. Für die Beantwortung der Fragestellungen war kein Zeitlimit gesetzt.

Im Anschluss wurde ein Rechner mit der Applikation vor der Testperson positioniert. Die Lage des Roboters, der Aufnahmebereich, sowie der restliche Ablauf des Experiments blieben unverändert. Die Aufgabe wurde unmittelbar danach fortgesetzt.

Nachdem die Aufgabe beendet wurde, sollte der Benutzer einen Fragebogen zur Roboterwartung, seiner Beanspruchung während des Experiments, Akzeptanz und Bedienung der Applikation beantworten.

---

#### 5.2.4 Auswertung

---

Die Antworten auf die geschlossenen Aufgaben wurden mit einer Likert-Skala, basierend auf fünf Antwortmöglichkeiten von 1 (trifft voll zu) bis 5 (trifft überhaupt nicht zu), ausgewertet. Bei der Analyse der offenen Fragestellungen, wurde jede richtig beantwortete Frage mit 1 gewichtet. War die Antwort falsch oder nicht gegeben, betrug die Gewichtung 0.

Alle Teilnehmer haben angegeben, dass sie routiniert im Umgang mit dem PC sind. 78.9% kennen oder haben schon Erfahrungen mit Augmented Reality gesammelt. 21% der Befragten verfügen über ein geringes Wissen auf dem Gebiet der Robotik. 13 von 19 Personen haben schon mit oder in der Nähe eines Roboters gearbeitet, davon sind 61.6% routiniert im Umgang mit dem Roboter und 84.7% haben sich in der Nähe des Roboters wohl gefühlt.

In der Abbildung 5.11 ist die Verteilung der richtigen Antworten auf offene Fragestellungen von und nach der Unterbrechung des Experiments, dargestellt.

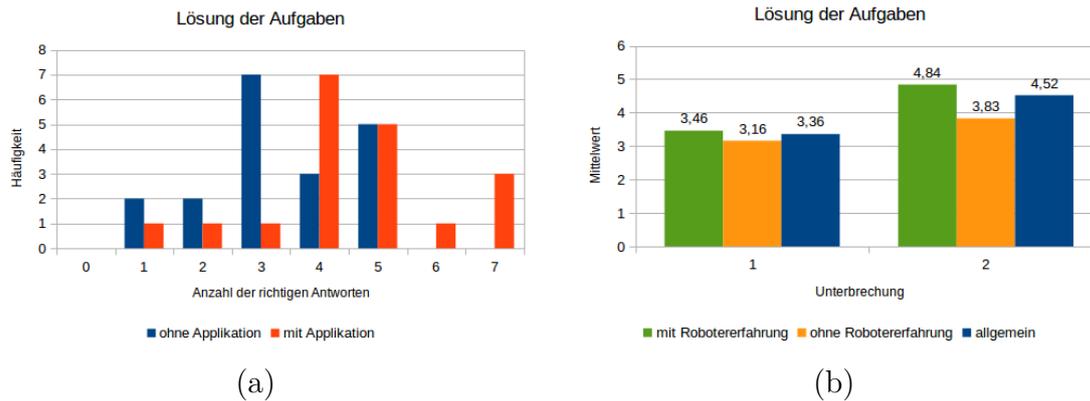


Abbildung 5.11: Verteilung der richtigen Antworten während des Experiments. (a) Im Schnitt wurden 3.36 Fragen, nach der ersten und 4.57 nach der zweiten Unterbrechung richtig beantwortet. Die Personen mit Robotererfahrung haben grundsätzlich bessere Ergebnisse, als die anderen erzielen können. (b) Anhand von der Häufigkeitsverteilung sieht man eine signifikante Änderung der Fähigkeiten der Testpersonen, im Bezug auf die Anzahl der richtigen Antworten. Diese Ergebnisse deuten auf den positiven Einfluss der in der Applikation dargestellten Informationen auf das Situationsbewusstsein

Nach dem ersten Abschnitt des Experiments wurden im Schnitt 3.36 Fragen richtig beantwortet. Nach der zweiten Unterbrechung sind es im Mittel 4.57 Fragen gewesen. Obwohl sich diese Ergebnisse nicht drastisch voneinander unterscheiden, sieht man anhand von der Häufigkeitsverteilung (vgl. Abbildung 5.11(a)) eine signifikante Änderung der Fähigkeiten, im Bezug auf die Anzahl der richtigen Antworten, die während beiden Experiment Abschnitten erteilt wurden. Vergleicht man die durchschnittlich erzeugten Ergebnisse der Probanden, die mit einem Roboter schon gearbeitet haben, im Vergleich zu den unerfahrenen, so wird deutlich, dass die vorhandene Robotererfahrung bei erneuter Kollaboration Vorteile mit sich bringt.

Diese Ergebnisse legen die Hypothese nahe, dass dank der zusätzlichen Informationen die Testpersonen ein besseres Situationsverständnis erlangten und eine bessere Prognose bezüglich des späteren Roboterhaltens geben konnten. Wobei die Präzision der Prognose mit der Erfahrung im Umgang mit dem Roboter gesteigert wird. Mit anderen Worten: Die verwendeten Anzeigen haben einen positiven Einfluss auf das Situationsbewusstsein.

Die Wahrnehmung des aktuellen Roboterzustands ist durch die Angabe des Zustimmungsgades auf folgende Feststellungen beurteilt worden:

- i Ich hatte Schwierigkeiten die Gefahrensituationen (z.B. Kollisionen) zu erkennen.
- ii Ich hatte Schwierigkeiten die Intention des Roboters (z.B. Richtung der Bewegung) zu erkennen.
- iii Ich konnte leicht die Bewegung des Roboters nachvollziehen.

Die Analyse dieser Daten zeigt, dass die Probanden bei der Nutzung der Applikation ein besseres Verständnis für das Verhalten des Roboters erlangten. Es ermöglichte dem Benutzer die Richtung der Bewegung des Roboters besser nachzuvollziehen und die Gefahrensituationen sowie die Intention zu erkennen (vgl. Abbildung 5.12). Bei der Bewertung der zweiten Frage haben drei Personen angegeben, dass sie bei der Verwendung der Applikation, die Gefahren nicht erkennen konnten. Eine der Ursachen könnte persönliches räumliches Vorstellungsvermögen sein. Durch die aus Sicht des Probanden falsch ausgewählte Perspektive der Kamera kann der die räumliche Lage des Roboterarmes oder der Objekte auf dem Tisch nicht erkennen.

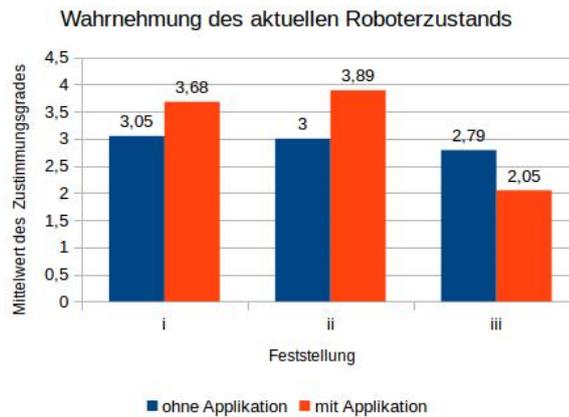


Abbildung 5.12: Bewertung des aktuellen Roboterzustands. Die zu bewertete Feststellungen: i) Ich hatte Schwierigkeiten die Gefahrensituationen (z.B. Kollisionen) zu erkennen, ii) Ich hatte Schwierigkeiten die Intention des Roboters (z.B. Richtung der Bewegung) zu erkennen iii) Ich konnte leicht die Bewegung des Roboters nachvollziehen. Die Auswertung der erhobenen Daten bestätigt, dass die Wahrnehmung des aktuellen Roboterzustands ausgeprägter ist, falls die Applikation benutzt wurde.

Die Beurteilung der Wahrnehmung des allgemeinen Roboterzustands war nur bei der Nutzung der Applikation möglich. D.h. während der zweiten Unterbrechung wurden Fragen zur Wartung und der Leistung des Roboters gestellt.

	ohne Robotererfahrung	mit Robotererfahrung	Allgemein
Genauigkeit ist schlechter geworden	83.3%	84.6%	84.2%
Roboter muss gewartet werden	50%	92.3%	78.9%
Roboter ist in einem guten Zustand	0%	61.5%	47.3%
Häufigkeit der Prozessfehler ist mit der Zeit gewachsen	83.3%	84%	78.9%

Tabelle 5.1: Einschätzung des Roboterzustands unter zu Zuhilfenahme der Applikation.

Durchschnittlich konnten über 61% der erfahrenen Probanden, im Gegensatz zu den anderen, den Zustand des Roboters korrekt bewerten. Die unerfahrenen Personen haben den Zustand des Roboters auf „zu gut“ im Vergleich zur tatsächlichen Zustand geschätzt. Diese Wirkung lässt sich eventuell mit dem während des Experiments dargestellten prozentualen Leistungsfähigkeit des Roboters, die 60% betrug, erklären. Im Allgemeinen wird diese Prozentzahl positiv wahrgenommen, jedoch ist es für den Zustand der Maschine schon kritisch. Von diesem Fakt abgesehen, hat sich herausgestellt, dass die verwendeten Darstellungsformen, die korrekte Bewertung der Leistung des Robotersystems unterstützen.

Insgesamt haben sich 89% der Experimentteilnehmer bei der Lösung der Aufgabe mit der Benutzung der Applikation sicherer oder gleich sicher (als ohne Applikation) und 68% wohler oder gleich wohl (ohne Applikation) gefühlt. Wie erwartet haben sich Personen, die mit dem Roboter schon gearbeitet haben, im Durchschnitt nicht sicherer mit der Applikation als ohne gefühlt. Dies hängt mit der Tatsache zusammen, dass diesen Personen den Roboter und sein sicheres Verhalten schon bekannt war. Bei anderen Personen ist das Sicherheitsgefühl signifikant gestiegen.

Aus diesen Ergebnissen kann die folgende Aussage abgeleitet werden: Der dargestellte Ansatz der visuellen Darstellung des Roboterzustand und -prozesses hat keinen negativen Einfluss auf die Empfindung des Sicherheits- und des Wohlfühls des Menschen. Zusätzlich hat, über die Hilfe (57.9% ) der Probanden angegeben, dass sie dem Roboter bei der Applikation Nutzung mehr vertrauen.

Alle Testpersonen haben angegeben, dass sie die Applikation noch einmal benutzen würden und, dass sich der Komfort in Bezug auf die Zusammenarbeit mit dem BioRob-Arm erhöht hat. Dabei finden 73.7% der Befragten, dass die Arbeit mit dem Roboter und die Benutzung der Applikation ihre Sicherheit erhöht.

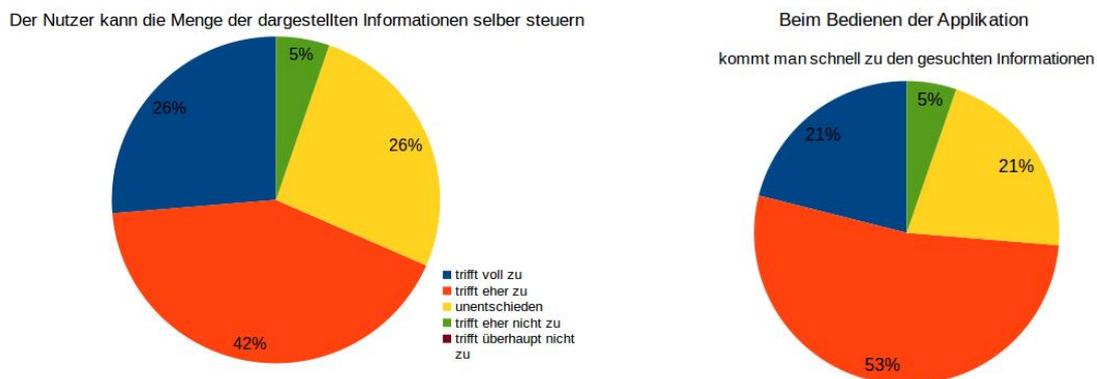


Abbildung 5.13: Bewertung der Benutzeroberfläche. Die meisten Testpersonen haben die Benutzeroberfläche der Applikation positiv bewertet. Sie finden, dass sie dem Nutzer, die Möglichkeit gibt die Menge der angezeigten Informationen selbst zu steuern und schnell zu den gesuchten Daten führt.

Die Mehrheit der Probanden ist der Meinung, dass die dargestellten Informationen ausreichend sind, um den Zustand und Prozess des Roboters zu verstehen. Die meisten

---

haben angegeben, dass die Struktur der graphischen Oberfläche dem Anwender beim Finden der gesuchten Daten unterstützt.

Die Auswertung der Frage auf welche Bereiche der Applikation die Person am meisten geachtet hat, haben die erwarteten Effekte gezeigt. Die meisten Personen haben ihre Aufmerksamkeit auf den Livestream mit den Augmented Reality Elementen gelenkt. An der zweiten Stelle lag der Bereich Movement. Vermutlich hängt diese Tatsache mit den gestellten Aufgaben zusammen. Zum Beispiel beinhaltet der Fragebogen keine Fragestellungen bezüglich der exakten Position des Endeffektors, oder den Verlauf des Prozesses.

Zur Analyse der während der Studie erhobenen Daten wurde SPSS Statistics (IBM) und LibreOffice Calc verwendet.

---

### 5.3 Diskussion

---

Wie die Studie gezeigt hat, führt die Darstellung des Roboterzustands und -prozesses zu einer Unterstützung des Situationsbewusstseins vom Anwender. Dies gilt vor allem bei der Verwendung der Augmented Reality, was mit der menschlichen Wahrnehmungspsychologie (der Mensch reagiert auf bewegliches immer mehr als auf statische Elemente) zusammenhängt.

Augmented Reality Elemente erlauben dem Anwender die Endeffektortrajektorie, den Sicherheitsraum, sowie den Start- und Zielbereich des Roboterprozesses zu visualisieren. Durch die Bewegung der Kamera kann der Benutzer diese Informationen aus verschiedenen Perspektiven betrachten. Es hat sich jedoch herausgestellt, dass eine falsch gewählte Kameraperspektive den Benutzer bei der Erkennung von Gefahren hindern kann. Die 3D Anzeige hingegen bietet, einen Vorteil, wenn der Mensch den Roboter aus verschiedenen Perspektiven, ohne die eigene Position zu verändern, betrachten will.

Bezüglich der Roboterinspektion wurde beobachtet, dass komplexe Roboterprozesse mit Hilfe von einer eindimensionalen Elementenliste dargestellt werden können, wodurch sie auch für die ungeübten Anwender verständlich sind.

Durch die Angaben bezüglich des Verlaufs der Wartungsmessungen, wie Bewegungsgenauigkeit, Betriebs-, oder Prozessausführungszeiten, kann der allgemeine Roboterzustand richtig wahrgenommen werden. Der Benutzer kann die negativen Trends schneller entdecken und entsprechende Gegenmaßnahmen einleiten. Allerdings, wie sich schon in der Benutzerstudie gezeigt hat, sind, im Gegensatz zu Toleranzniveaus, die prozentualen Angaben der Funktionsfähigkeit ungeeignet.

Obwohl nicht alle dargestellten Roboterzustände und -prozessvariablen stets wahrgenommen werden, hilft die benutzerfreundliche Darstellung dem Anwender das bisher unvorhersehbare Verhalten des Roboter zu verstehen und nachzuvollziehen, wodurch er sich in der Nähe des Roboters sicherer und komfortabler fühlt.

Bei der Implementierung der graphischen Schnittstelle stand die Benutzerfreundlichkeit im Vordergrund. Deshalb wurden die Vorgaben zur ergonomischen Anwendungsgestaltung (siehe 2.3) angewendet. Die entwickelte Benutzeroberfläche kann durch das Ein- und Ausblenden der einzelnen Bereiche an die Anwenderbedürfnisse und -erfahrungen angepasst

---

werden, so dass nur relevante Informationen angezeigt werden. Zusätzlich werden die semantisch zusammenhängende Daten gruppiert, wodurch die Navigation vereinfacht wird. Mit dem Hervorheben der Variablen wird die Aufmerksamkeit des Benutzers auf aktuelle Ereignisse gelenkt. Während der Benutzerstudie konnte die Benutzeroberfläche im Hinblick auf die Effizienz, Benutzbarkeit und Bedienbarkeit höchste Bewertungen erzielen.

Durch die Verwendung von einer verteilten Architektur bei der Realisierung des Systems konnte eine effiziente Berechnung und Darstellung realisiert, sowie die Ressourcen der einzelnen Komponente geschont werden. Dabei ist die Systemeinheit für die Bestimmung und Transport der zur darstellenden Roboterdaten verantwortlich. Die Klientkomponente ist mittels Web Technologien realisiert worden, und somit durch gängige Betriebssysteme unterstützt. Sie ist verantwortlich für die Verwaltung der Benutzerinteraktionen und die graphische Darstellung der Roboterdaten.

Die im System erhaltene Kameraschnittstelle unterstützt sowohl einfache Web-, wie auch hochauflösende Industriekameras. Mit der Verwendung von *Ros*-Middlesoftware wurde eine universelle Schnittstelle zur Kommunikation mit einem Robotersystem entwickelt.

Die Verbindung zwischen den einzelnen Systemkomponenten (Server, Klient) basiert auf einem Websocket, was die Existenz einer Netzwerkverbindung voraussetzt. Diese muss über genügend Bandbreite verfügen, um ein stabiles Datentransport zu ermöglichen und so eine zuverlässige Visualisierung zu garantieren. Standardmäßig wird der Kameralivestream mit der Frequenz von 30 Frames pro Sekunde an den die Serverkomponente übermittelt, jedoch kann diese über die Benutzerschnittstelle an die verfügbare Netzwerkbandbreite angepasst werden. Zu beachten ist, dass mit der geringen Frequenz zu Verzögerungen und Ausfällen in der Darstellung der Informationen kommen kann.

Die Genauigkeit mit welcher die virtuellen Objekte in die reale Umgebung eingeblendet werden hängt unmittelbar von der Präzision des angewendeten ARToolkit Markertrackingverfahren [49] [61] [62] und der Sorgfalt mit welcher das *urdf*-Modell des verwendeten Roboters aufgestellt wurde. Dadurch ist sie sehr Robotersystem bezogen und dessen Bewertung nicht, als Teil dieser Arbeit angesehen wird.

#### **Zusammenfassend:**

Für den Anwender bedeutet die Verwendung der Applikation, dass die Arbeit mit oder in der Nähe eines Roboters angenehmer und sicherer gestaltet werden kann, falls zusätzliche Informationen über das Vorhaben des Roboterarmes zur Verfügung stehen. Durch die Visualisierung der Wartungsinformationen können Rückschlüsse auf die zukünftigen Ausfälle, zu erwartende Qualität der Ausführung und Änderungen der Leistung gezogen werden.

Die Akzeptanz des Roboters und eine direkte Interaktion mit ihm wird dank der entwickelten Applikation signifikant gesteigert.

Durch Verwendung von allgemein gebräuchlichen Darstellungsformen ist die gewählte Projektion des Roboterzustands und -prozesses auch für unerfahrene Benutzer verständlich. Dank der gewählten Klient-Server-Architektur ist das System flexibel einsetzbar und erweiterbar. Über die entwickelten Roboter-Schnittstellen ist die Anwendung mit allen *Ros*-Robotersystemen kompatibel.

---

## 6 Zusammenfassung

---

In dieser Thesis wurde ein Konzept und dessen Realisierung zur Darstellung des Roboterzustands und -prozesses in der industriellen Umgebung, welches zum einen zur Verstärkung des Situationsbewusstseins, des mentalen Komforts und der Sicherheit des Anwenders, zum anderen zur erhöhten Produktivität und vereinfachter Überwachung der Automatisierung beiträgt, vorgestellt.

Es wurden die existierenden Ansätze zur Übermittlung grundlegender Roboterinformationen an den Menschen diskutiert. Dabei spielte die Anwendbarkeit in der industriellen Automatisierung eine Rolle. In vielen Ansätzen wurde der Anwender nicht über den Ablauf und das Ergebnis des Roboterprozesses informiert. Einige Lösungsansätze benötigten zur Informationsübermittlung neben dem Roboter selbst, zusätzliche Sensoren oder Vorrichtungen. Dies bot eine potentiell einfachere und vor allem eine intuitivere Lösung zu suchen. Aus diesem Grund wurde ein Roboter-Kamera System entwickelt, das Augmented Reality zur Einblendung von Informationen direkt in die Umgebung des Roboters mit Hilfe eines Video-See-Though Ansatzes, verwendet.

Der entwickelte Ansatz ist an den Experten, sowie unerfahrene Nutzer adressiert, die mit dem Roboter oder in seiner Nähe arbeiten. So kann der Arbeiter ein mobiles Endgerät in der Hand halten und aus seiner eigenen Perspektive den für ihn bisher unbekanntem Verlauf der Roboterbewegung, den Sicherheitsbereich, oder den Prozesszustand betrachten. Die Wartungsinformationen helfen dem Benutzer dabei den allgemeinen Zustand des Roboters zu bewerten. Dadurch kann möglichen Ausfällen und Abweichungen in der Prozessausführung vorgebeugt werden.

Zur Realisierung des Zielsystems musste zunächst ein Verfahren zur Kamera-Roboter Kalibrierung entwickelt werden. Das vorgestellte Kalibrierungsverfahren benötigt lediglich eine Kamera und mehrere Marker (mindestens einen), die einfach auf Papier gedruckt werden können. Diese werden zur Bestimmung der Relation zwischen Kamera und Roboter benötigt, welche die Grundlage einer positionsgenauen Überlagerung der realen Objekte mit dem virtuellen Inhalten bildet, benötigt. Das entwickelte Verfahren zur Kameralokalisierung, basiert auf Markertracking. Es gestattet die Positionierung beliebig vieler Marker auf vom Benutzer gewählten Roboterglied gleichzeitige Änderungen der Lage des Aufnahmegäräts und die Bewegung des Roboterarmes (was bei den üblichen AR Systemen nicht gegeben ist).

Nachdem das Verfahren zur Kamera-Roboter-Lokalisierung und damit die Grundvoraussetzung zur Realisierung einer AR basierten Benutzerschnittstelle entwickelt wurde, mussten die darzustellenden Roboter- und Prozessinformationen definiert werden, welche die Server-Klient Architektur des Systems, sowie eine graphische Mensch-Maschine-Schnittstelle erstellt.

Die realisierte Software zeichnet sich durch ihre flexible Struktur in Hinblick auf folgende Komponenten aus: dem Roboter (der lediglich über eine *Ros* Schnittstelle und ein *urdf*-Modell verfügen muss), die Einsatzumgebung (es werden ausschließlich eine Kamera

---

und Marker benötigt), sowie die Klientkomponente (welche dank der Realisierung mittels Webtechnologie, durch alle gängigen Betriebssysteme unterstützt wird). Eine wichtige Voraussetzung ist, dass die einzelnen Systemkomponenten (Server und Klient) über eine Netzwerkverbindung verfügen, welche die notwendige Bandbreite besitzen um eine effiziente und zuverlässige Kommunikation zu ermöglichen.

Da kein Fachwissen zur Nutzung der entwickelten Applikation benötigt werden sollte, ist ein besonderer Fokus auf die Entwicklung und Realisierung einer intuitiven und leicht bedienbaren Benutzeroberfläche gelegt worden. Diese wurde in vier semantisch zusammenhängende Bereiche (Status, aktueller Zustand Prozess und Wartung, Logging) aufgeteilt, was dem Benutzer einen schnellen Überblick, sowohl über die dargestellten Informationen, als auch über die gesamte Funktionalität der graphischer Benutzeroberfläche bieten soll.

Nachdem das System zur Visualisierung des Roboterzustandes und -prozesses erstellt wurde, konnte das Verfahren, in Bezug auf die Darstellung am konkreten Beispiel eines Leichtbauroboters, evaluiert und diskutiert werden. Zusätzlich wurde eine Benutzerstudie durchgeführt, welche den Einfluss der in der Benutzeroberfläche dargestellten Informationen auf den Benutzer, beurteilt. Durch diese Studie konnte gezeigt werden, dass der Roboter- sowie Prozesszustand an unerfahrene Anwender übermittelt werden konnte, das verwendete Konzept eine positive Wirkung auf die Roboter-Mensch Kommunikation hat und zur Steigerung des Situationsverständnisses des Anwenders beiträgt. Darüber hinaus stellte sich das entwickelte System als flexibel einsetzbar heraus. Dank dem angewendeten Kalibrierungsverfahren und der Existenz von mehreren Marker wurde eine zuverlässige und effiziente Visualisierung der Roboter- und Prozessinformationen möglich und der Aufwand der Systemintegration minimiert.

Der in dieser Arbeit vorgestellte Ansatz zur Roboterzustand und -prozessvisualisierung in der industriellen Automatisierungsumgebung kann als Grundlage für weiterführende Arbeiten verstanden werden. So kann zum Beispiel das vorgestellte Visualisierungskonzept auf ein Head-mounted display (HMD), oder eine Datenbrille übertragen werden. Zurzeit muss der Benutzer das mobile Endgerät (z.B. Tablett-Computer oder Handy), bei der Interaktion mit dem entwickelten System in einer, oder beiden Händen halten, wodurch er während der Ausführung seiner Aufgabe beschränkt wird. Um dies zu vermeiden, könnten die HMD und AR Technologien miteinander kombiniert werden.

Neben den in der entwickelten Applikation dargestellten Informationen, ist die Visualisierung mit Hilfe von AR (während der Studie der Livestream am meisten beachtet wurde) weiterer Prozessinformationen denkbar. Diese müssen jedoch mit Bedacht ausgewählt werden, da es sonst durch die Überlagerung des Streams zu Verwirrungen kommen kann.

Als Erweiterung kann der vorgestellte Ansatz um Serviceinformationen ergänzt werden, welche bei der Reparatur oder Wartung des Roboters visuelle Reparaturanweisungen und benötigte Werkzeuge in die Roboterumgebung darstellt.

---

## Literaturverzeichnis

---

- [1] BMW Group. Mini augmented vision: A revolutionary display concept offering enhanced comfort and safety. exclusive prototype of augmented reality eyewear underlines the innovative flair and creativity of the mini brand. <https://www.press.bmwgroup.com/global/article/detail/T0212042EN/mini-augmented-vision:-a-revolutionary-display-concept-offering-enhanced-comfort-and-safety-exclusive-prototype-of-augmented-reality-eyewear-underlines-the-innovative-flair-and-creativity-of-the-mini-brand?language=en>, 2016-08-16.
- [2] L. Maloney. A new face for robotics. <http://insights.globalspec.com/article/20/a-new-face-for-robotics>, 2016-08-16.
- [3] Bionic Robotics GmbH. Bionic robotics. <http://www.bionic-robotics.de>, 2016-08-16.
- [4] IFR. World robotics 2015 industrial robots. Technical report, International Federation of Robotics (IFR), 2015.
- [5] M. Huelke. Arbeitsplätze der industrie 4.0 - kollaborierende roboter. *DGUV Forum*, (3):10–13, 2015.
- [6] M. Schenk and N. Elkmann. Sichere mensch-roboter-interaktion: Anforderungen, voraussetzungen, szenarien und lösungsansätze. *Demografischer Wandel-Herausforderungen für die arbeits-und Betriebsorganisation der Zukunft*, pages 109–120, 2012.
- [7] ISO. Iso ts 15066:2016: Robots and robotic devices – collaborative robots, 2016.
- [8] ISO. *ISO 10218-1 (2012)*. International Organization for Standardization.
- [9] ISO. *ISO 10218-2 (2011): Robots and robotic devices - Safety requirements for industrial robots - Part 2: Robot systems and integration*. International Organization for Standardization, 2011.
- [10] Berufsgenossenschaft Holz und Metall. Dguv information 209-074: Industrieroboter, Jan 2015.
- [11] S. Green, Geoffrey Ch., and M. Billinghamurst. *Augmented Reality for Human-Robot Collaboration*. INTECH Open Access Publisher, 2007.
- [12] M. Endsley. Toward a theory of situation awareness in dynamic systems. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 37(1):32–64, 1995.
- [13] R. Nispeling. Optimize super (it) robot condition monitoring tool. *ABB Review*, (2):40–47, 2002.

- 
- [14] R. Dörner, W. Broll, and P. Grimm. *Virtual und Augmented Reality (VR/AR). Grundlagen und Methoden der Virtuellen und Augmentierten Realität*. Springer Vieweg, 2013.
- [15] A. Mehler-Bicher and L. Steiger. *Augmented Reality: Theorie und Praxis*. Walter de Gruyter GmbH & Co KG, 2014.
- [16] R. Azuma. A survey of augmented reality. *Presence: Teleoperators and virtual environments*, 6(4):355–385, 1997.
- [17] J. Rolland, L. Davis, and Y. Baillot. A survey of tracking technology for virtual environments. *Fundamentals of wearable computers and augmented reality*, 1:67–112, 2001.
- [18] H. Rosch. Augmented reality, eine übersicht. Technical report, Technische Universität Ilmenau Fachgruppe Graphische Datenverarbeitung, 1999.
- [19] H. Belghit and A. Bellarbi. Object recognition based on orb descriptor for markerless augmented reality.
- [20] BMW Group. Kothes! augmented documentation. <https://www.reflekt.com/newsroom/>, 2016-08-16.
- [21] Situationsbewusstsein. <http://macs2.psychologie.hu-berlin.de/aio/index.php/mensch-in-aktion/aufmerksamkeit/grundlagen-aufmerksamkeit/116-situationsbewusstsein>, 2016-08-16.
- [22] D.G. Endsley, M. R. Jones. *Designing for Situation Awareness: An Approach to User-Centered Design*. CRC Press, 2011.
- [23] ENISO DIN. 9241-110: Ergonomie der mensch-system-interaktion–teil 110: Grundsätze der dialoggestaltung. *Berlin: Beuth*, 2006.
- [24] D. Zühlke. *Nutzergerechte Entwicklung von Mensch-Maschine-Systemen: Useware-Engineering für technische Systeme*. Springer-Verlag, 2011.
- [25] E. Gamma, R. Johnson, R. Helm, and J. Vlissides. *Entwurfsmuster: Elemente wiederverwendbarer objektorientierter Software*. Pearson Deutschland GmbH, 2011.
- [26] ENISO DIN. 14915-1: Software ergonomie für multimedia-benutzerschnittstellen. teil 1: Gestaltungsgrundsätze und rahmenbedingungen. *Berlin: Beuth*, 2003.
- [27] A. Heinecke. *Mensch-Computer-Interaktion: Basiswissen für Entwickler und Gestalter*. Springer-Verlag, 2011.
- [28] M. Endsley. Theoretical underpinnings of situation awareness: A critical review. *Situation awareness analysis and measurement*, pages 3–32, 2000.
- [29] M. Goodrich and A. Schultz. Human-robot interaction: a survey. *Foundations and trends in human-computer interaction*, 1(3):203–275, 2007.

- 
- [30] K. Kobayashi and S. Yamada. Informing a user of robot's mind by motion. In *The Third International Conference on Computational Intelligence, Robotics and Autonomous Systems (CIRAS 2005), SS4B-3*, 2005.
- [31] K. Kobayashi and S. Yamada. *Making a Mobile Robot to Express its Mind by Motion Overlap*. INTECH Open Access Publisher, 2009.
- [32] T. Komatsu. Can we assign attitudes to a computer based on its beeps?-toward an effective method for making humans empathize with artificial agents. In *INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE*, volume 19, page 1692. LAWRENCE ERLBAUM ASSOCIATES LTD, 2005.
- [33] Ch. Vogel. Experimental evaluation of advanced sensor-based supervision and work cell integration strategies - excell.
- [34] Fraunhofer-Institut für Fabrikbetrieb und Automatisierung IFF. Optische arbeitsraumüberwachung. <http://www.iff.fraunhofer.de/de/geschaeftsbereiche/robotersysteme/echord-excell.html>, 2016-08-16.
- [35] N. Elkmann, R. Behrens, M. Fritzsche, and Ch. Vogel. Neue schutzkonzepte für das 'friedliche' miteinander. <http://www.computer-automation.de/feldebene/sensoren/artikel/95520/>, 2016-08-16.
- [36] Rethink Robotics. Baxter, 2016. <http://www.rethinkrobotics.com/de/baxter/>, 2016-08-16.
- [37] R. Chadalavada, H. Andreasson, R. Krug, and A. Lilienthal. That's on my mind! robot to human intention communication through on-board projection on shared floor space. In *2015 European Conference on Mobile Robots (ECMR)*. Institute of Electrical & Electronics Engineers (IEEE), sep 2015.
- [38] R. Chadalavada, H. Andreasson, R. Krug, and A. Lilienthal. Empirical evaluation of human trust in an expressive mobile robot. 2016.
- [39] O. Bimber and R. Raskar. *Spatial augmented reality: merging real and virtual worlds*. CRC press, 2005.
- [40] Ros, aug 2016.
- [41] Anis Koubaa, editor. *Robot Operating System (ROS)*. Springer International Publishing, 2016.
- [42] urdf, aug 2016.
- [43] J. Lentin. *Mastering ROS for Robotics Programming -*. Packt Publishing Ltd, 1. Aufl. edition, 2015.

- 
- [44] T. Foote. tf: The transform library. In *Technologies for Practical Robot Applications (TePRA), 2013 IEEE International Conference on*, pages 1–6. IEEE, 2013.
- [45] J. Lee. Web applications for robots using rosbridge. *Brown University*, 2012.
- [46] Crick Ch., J. Graylin, and Osentosiki S. Rosbridge: Ros for non-ros users. 2011.
- [47] Alexander B., Hsiao K., and Jenkins Ch. Robot web tools. *IEEE ROBOTICS & AUTOMATION MAGAZINE*, 2012.
- [48] Artoolkit, aug 2016. <http://www.artoolkit.org>, 2016-08-16.
- [49] H. Kato and M. Billinghurst. Marker tracking and hmd calibration for a video-based augmented reality conferencing system. In *Augmented Reality, 1999. (IWAR'99) Proceedings. 2nd IEEE and ACM International Workshop on*, pages 85–94. IEEE, 1999.
- [50] Opencv, aug 2016. <http://opencv.com/>, 2016-08-16.
- [51] R. Laganière. *OpenCV 2 Computer Vision Application Programming Cookbook*. Packt Pub Limited, 2011.
- [52] Hans-Jürgen Siegert and Siegfried Bocionek. *Robotik: Programmierung intelligenter Roboter: Programmierung intelligenter Roboter*. Springer-Verlag, 2013.
- [53] John J Craig. *Introduction to robotics: mechanics and control*, volume 3. Pearson Prentice Hall Upper Saddle River, 2005.
- [54] Simon JD Prince. *Computer vision: models, learning, and inference*. Cambridge University Press, 2012.
- [55] Herbert Süße and Erik Rodner. *Bildverarbeitung und Objekterkennung*. Springer Science + Business Media, 2014.
- [56] ABB. Abb-technologien, die die welt veränderten, 2010.
- [57] O. von Stryk. Robotik 1 - skriptum, 2011.
- [58] Günther Bengel, Christian Baum, Marcel Kunze, and Karl-Uwe Stucky. Masterkurs parallele und verteilte systeme. *Vieweg+ Teubner Verlag*, 2008.
- [59] Igor R Belousov, Ryad Chellali, and Gordon J Clapworthy. Virtual reality tools for internet robotics. In *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*, volume 2, pages 1878–1883. IEEE, 2001.
- [60] ARToolKit. Tutorial 3: Multi-marker tracking, 2016. <https://www.hitl.washington.edu/artoolkit/documentation/tutorialmulti.htm>, 2016-08-16.
- [61] I. Rabbi, S. Ullah, and A. Alam. Marker based tracking in augmented reality applications using artoolkit: A case study.

- 
- [62] D. Abawi, J. Bienwald, and R. Dorner. Accuracy in optical tracking with fiducial markers: An accuracy function for artoolkit. In *Proceedings of the 3rd IEEE/ACM International Symposium on Mixed and Augmented Reality, ISMAR '04*, pages 260–261. IEEE Computer Society, 2004.