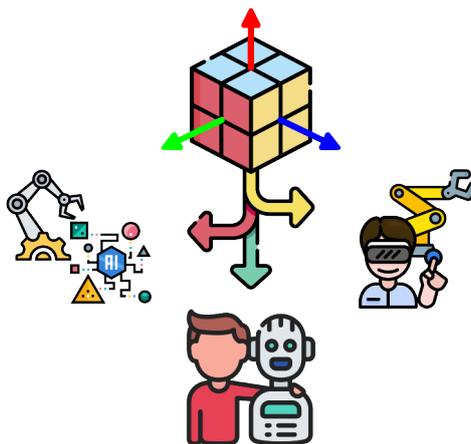


Institut für Montagetechnik und Industrierobotik

Prof. Dr.-Ing A. Ratz

Masterarbeit

6D-Posenbestimmung von neuartigen Objekten für AR-basierte Programmierung durch Demonstration mit dem HRC-Hub



Verfasser

David Wendorff

Matrikel-Nr: 2881480

Match-Kennung: MA 02/24

Betreuer:

Sebastian Blankemeyer, M.Sc

Garbsen, den 14.08.2024

Masterarbeit

Aufgabenstellung für Herrn David Wendorff

Studiengang: Master Produktion und Logistik

Matr. Nr.: 2881480

Titel: 6D-Posenbestimmung von neuartigen Objekten für AR-basierte Programmierung durch Demonstration mit dem HRC-Hub

Titel (engl.): 6D Pose Estimation of Novel Objects for AR-Based Programming by Demonstration with the HRC-Hub

Allgemeines:

Produzierende Unternehmen, insbesondere solche mit einem hohen Anteil manueller Prozesse, stehen vor einem komplexen Dilemma. Einerseits ist der Einsatz von Automatisierung für die Unternehmen entscheidend, um wettbewerbsfähig zu bleiben. Dies gilt insbesondere in Hochlohnländern, in denen der Fachkräftemangel, z.B. aufgrund der Alterung der Bevölkerung, ein zunehmendes Problem darstellt. Andererseits ist es erforderlich, eine hohe Flexibilität in der Produktion zu gewährleisten, um individuelle Kundenwünsche zu erfüllen und schnell auf Veränderungen reagieren zu können. Leichtbauroboter (LBR), die mit Menschen kollaborieren, sind anpassungsfähige Automatisierungssysteme mit großem Potenzial, um diesen Herausforderungen zu begegnen. Die Augmented Reality (AR) bietet hierbei eine geeignete Schnittstelle, um die nutzerfreundliche Programmierung durch Demonstration in der Mensch-Roboter-Kollaboration (MRK) zu ermöglichen. Eine robuste und flexible Posenbestimmung ist dabei unerlässlich, um die AR sinnvoll und zielführend einzusetzen. Diese ermöglicht die Bestimmung der Position und Orientierung von Objekten in der realen Welt, welche anschließend in Programmabläufe überführt werden können. Allerdings müssen sich die Anwender hierbei an unterschiedliche Herstellerformate gewöhnen und es fehlt oft ein direktes Nutzerfeedback. Darüber hinaus stellt die Integration der jeweiligen Roboter in die unternehmensspezifische Produktionsumgebung eine Herausforderung dar, da eine Vielzahl unterschiedlicher und meist proprietärer Schnittstellen genutzt werden muss.

Aufgabenstellung:

Das Ziel dieser Arbeit ist die Weiterentwicklung eines ganzheitlichen Ansatzes für die herstellerunabhängige Automatisierung mithilfe der Mensch-Roboter-Kollaboration, des Human-Robot Collaboration Hub (HRC-Hub). Zunächst soll der aktuelle Stand der Technik für die 6D-Posenbestimmung, insbesondere von neuartigen Objekten, recherchiert werden. Anschließend soll das MegaPose-Verfahren für die Posenbestimmung in der Augmented Reality implementiert werden. Anschließend soll das Verfahren hinsichtlich seiner erreichten Genauigkeit bewertet werden. Hierfür ist die Entwicklung eines Verfahrens erforderlich, das die Genauigkeit in einem automatisierten und robotergestützten Prozess bewertet. Das entwickelte Verfahren wird anschließend implementiert und zur Bewertung der

Posenbestimmung verwendet. Darüber hinaus müssen die Anforderungen an die einzelnen Bestandteile des HRC-Hubs identifiziert werden. Auf dieser Grundlage soll ein detailliertes Konzept für den HRC-Hub ausgearbeitet werden. Gegebenenfalls sollen weitere Komponenten konzipiert und entwickelt werden.

Im Rahmen dieser Arbeit ergeben sich insbesondere die folgenden Aufgabenpunkte:

- Recherche des aktuellen Standes der Technik bezüglich verschiedener Verfahren zur Pose Estimation
- Implementierung des MegaPose-Verfahrens zur Bestimmung der Pose vorgegebener Objekte anhand ihrer CAD-Modellen und unter Verwendung eines Intel RealSense Sensors
- Entwicklung eines automatisierten und robotergestützten Messverfahrens zur Bewertung der Genauigkeit der Posenbestimmung
- Implementierung des Messverfahrens mithilfe eines geeigneten Industrieroboters
- Bewertung der erreichten Genauigkeit der Posenbestimmung anhand eines vorgegebenen Anwendungsfalls
- Identifikation von Anforderungskriterien an einzelne Bestandteile des HRC-Hubs
- Ausarbeitung eines detaillierten Konzepts für den HRC-Hub, unter Berücksichtigung der Anforderungskriterien
- Optional Implementierung einzelner Komponenten des HRC-Hubs
- Schriftliche Dokumentation der Ergebnisse
- Präsentation der Arbeit

Die Bearbeitungszeit richtet sich nach der für den/die Studierende/n gültigen Prüfungsordnung.

Die Arbeit ist in engem Kontakt mit dem/der durch die Prüferin bestellten Betreuer/in durchzuführen. In regelmäßigen Abständen ist über den Fortgang der Arbeit zu berichten. Die Arbeit ist spätestens 6 Monate nach Ausgabe des Themas in zwei permanent gebundenen Exemplaren (Leim-, Thermo- oder Buchbindung) und auf einer CD/DVD bzw. USB-Stick am Institut abzugeben.

Betreuer/in: Sebastian Blankemeyer

match-Kennung: MA 02/24

Ausgabedatum: 14.02.2024

Abgabedatum: 14.08.2024

Prof. Dr.-Ing. Annika Raatz

David Wendorff

Erklärung

Hiermit versichere ich, dass ich die vorliegende Masterarbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe. Alle Passagen der Arbeit, die wörtlich oder sinngemäß aus anderen Quellen übernommen wurden, sind als solche kenntlich gemacht. Ich versichere darüber hinaus, dass die Arbeit in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorliegt.

Ort, Datum

Unterschrift

Kurzfassung

Die moderne industrielle Produktion steht vor zahlreichen Herausforderungen, insbesondere für Unternehmen mit einem hohen Anteil an manuellen Prozessen. Dies bringt einen verstärkten Druck zur Automatisierung mit sich, wobei gleichzeitig die notwendige Flexibilität gewahrt bleiben muss. In diesem Kontext gewinnt die Mensch-Roboter-Kollaboration (MRK) zunehmend an Bedeutung, da sie die Automatisierung und Produktivität deutlich steigern kann. Allerdings erfordert die Implementierung von MRK umfangreiche Programmierkenntnisse, die nicht immer vorhanden sind.

Hier bietet das Konzept des Programmierens durch Vormachen (PdV) eine innovative Lösung. PdV ermöglicht es Benutzern, Roboter zu programmieren, ohne tiefgehende Programmierkenntnisse haben zu müssen. Unterstützt durch neue Technologien wie Augmented Reality (AR), kann PdV besonders effektiv werden. Allerdings sind für eine erfolgreiche Umsetzung robuste Methoden zur 6D-Posebestimmung unerlässlich. Trotz der Fortschritte im Bereich des maschinellen Lernens fehlen immer noch umfassende quantitative Bewertungen und ausreichend verfügbare Datensätze.

Diese Arbeit hat das Ziel, die Eignung aktueller Methoden zur 6D-Posenbestimmung für die intuitive Programmierung mittels PdV zu evaluieren. Um dies zu erreichen, wird ein automatisiertes Verfahren zur Aufnahme annotierter Daten entwickelt, und experimentelle Versuche mit aktuellen Methoden werden durchgeführt. Die Arbeit gliedert sich in verschiedene Abschnitte: Zunächst werden die theoretischen Grundlagen dargelegt, gefolgt von der konkreten Formulierung der Forschungsziele. Anschließend wird die wissenschaftliche Vorgehensweise erläutert, bevor die praktische Umsetzung beschrieben wird. Die Analyse der Ergebnisse bildet einen weiteren Schwerpunkt, ehe die Arbeit mit einem Überblick über die erzielten Erkenntnisse und möglichen Perspektiven für zukünftige Forschungen abschließt.

Durch diese umfassende Herangehensweise soll ein Beitrag zur Optimierung der Mensch-Roboter-Kollaboration geleistet und eine Brücke zwischen Technologie und Anwendern ohne tiefgehende Programmierkenntnisse geschlagen werden.

Schlagerworte: Mensch-Roboter-Kollaboration (MRK), Programmieren durch Vormachen (PdV), Augmented Reality (AR), 6D-Posenbestimmung

Abstract

6D Pose Estimation of Novel Objects for AR-Based Programming by Demonstration with the HRC-Hub

Modern industrial production faces numerous challenges, particularly for companies with a high proportion of manual processes. This increases the pressure for automation while simultaneously maintaining the necessary flexibility. In this context, human-robot collaboration (HRC) is gaining increasing importance as it can significantly enhance automation and productivity. However, implementing HRC requires extensive programming knowledge, which is not always readily available.

The concept of Programming by Demonstration (PbD) offers an innovative solution to this dilemma. PbD enables users to program robots without requiring in-depth programming skills. Supported by new technologies such as Augmented Reality (AR), PbD can become particularly effective. However, robust methods for 6D pose estimation are essential for successful implementation. Despite advancements in the field of machine learning, there is still a lack of comprehensive quantitative evaluations and sufficiently available datasets.

This work aims to evaluate the suitability of current methods for 6D pose estimation for intuitive programming using PbD. To achieve this, an automated procedure for capturing annotated data will be developed, and experimental trials with state-of-the-art methods will be conducted. The work is structured into multiple sections: initially, the theoretical foundations are presented, followed by the specific formulation of research objectives. Subsequently, the scientific methodology is explained before detailing the practical implementation. The analysis of the results forms another focal point before the work concludes with an overview of the findings and potential perspectives for future research.

Through this comprehensive approach, this work aims to contribute to optimizing human-robot collaboration and bridging the gap between technology and end-users without extensive programming knowledge.

Keywords: Human-robot collaboration (HRC), Programming by demonstration (PbD), Augmented Reality (AR), 6D pose estimation

Inhaltsverzeichnis

Abbildungsverzeichnis	iii
Tabellenverzeichnis	vi
Quellcodeverzeichnis	vi
Abkürzungsverzeichnis	vii
1. Einleitung	1
2. Grundlagen und Stand der Technik	4
2.1. Mensch-Roboter-Kollaboration	4
2.2. Programmieren durch Vormachen	7
2.3. Beispiele aus der aktuellen Forschung	10
2.4. Grundlagen der 6D-Posenbestimmung	13
2.4.1. Bewertungsmetriken für die 6D-Posenbestimmung	16
2.4.2. Datensätze für die 6D-Posenbestimmung	20
2.4.3. Sensoren für die 6D-Posenbestimmung	27
2.5. Aktuelle Methoden für die 6D-Posenbestimmung von neuartigen Objekten	28
2.5.1. MegaPose	32
2.5.2. SAM6D	33
2.5.3. FoundationPose	35
2.6. HRC-HUB - ein holistisches Framework für die MRK	38
3. Zielsetzung der Arbeit	40
4. Methodik und konzeptionelle Gestaltung	41
4.1. Bilderverarbeitung	41
4.2. Berechnung der Genauigkeit	44
4.3. Theoretische Genauigkeit der Sensoren	48
4.4. Konzept und Versuchsplanung	51
5. Implementierung und praktische Durchführung	54
5.1. Praktischer Versuchsaufbau	54
5.1.1. Aufbau und Konfiguration der notwendigen Hardware	54
5.1.2. Roboterprogramm für die automatisierte Datenaufnahme	56

5.2.	Relative Festlegung der Ground-Truth	58
5.2.1.	Transformation der Koordinatensysteme mit pytransform3d	58
5.2.2.	Referenzierung der Sensoren über ArUco Marker	60
5.2.3.	Verwendung von Templates für die Ground-Truth	63
5.3.	Implementierung der Methoden zur 6D-Posenbestimmung	65
5.3.1.	Automatische Generierung der Objektmasken mit BlenderProc	65
5.3.2.	Angleichung der RGB-D Daten	66
5.3.3.	MegaPose	68
5.3.4.	SAM6D	69
5.3.5.	FoundationPose	71
5.4.	Übersicht über den Gesamtprozess	73
6.	Ergebnisse und Diskussion	75
6.1.	Vergleich der Methoden zur 6D-Posenbestimmung	75
6.2.	Vergleich der Bewegungen	78
6.3.	Vergleich der Sensoren	79
6.4.	Untersuchung des Einflusses der Sensorauflösung	81
6.5.	Untersuchung des Einflusses der Objektverdeckung	82
6.6.	Untersuchung des Einflusses der Bildfrequenz und Robotergeschwindigkeit bei der Aufnahme	84
6.7.	Validierung der Genauigkeit bei mehrfacher Wiederholung	86
6.8.	Fazit	87
7.	Zusammenfassung und Ausblick	88
8.	Literaturverzeichnis	91
A.	Anhang	99
A.1.	Objekte des ITODD Datensatzes	99
A.2.	Ergebnisse der verwendeten Methoden in der BOP Challenge	100
A.3.	Ergebnisse der Versuche	101
A.3.1.	Versuch A1	101
A.3.2.	Versuch A2	105
A.3.3.	Versuch B1	110
A.3.4.	Versuch B2	111
A.3.5.	Versuch C2	111
A.3.6.	Versuch C3	112

Abbildungsverzeichnis

1.	Kontext und Motivation der Arbeit	1
2.	Herleitung des Ziels der Arbeit	2
3.	Gliederung der Arbeit	3
4.	Entwicklung des Arbeitsraumes von sicherheits- zu kollaborationsorientiert	5
5.	Modalitäten und Techniken der Mensch-Roboter-Interaktion	6
6.	Ziele und Vorteile der visuellen Augmentation	6
7.	Einteilung der Roboterprogrammierung für die MRK in vier Ansätze und Unterscheidung der benutzerzentrierten-Programmierung	7
8.	Strukturierung von Programmieren durch Vormachen	7
9.	Verschiedene Arten von Sensoren für die Gestenerkennung	8
10.	Übersicht der Methode von Zou und Andulkar	10
11.	Verfahren und Versuchsaufbau der Methode von Blankemeyer u. a.	11
12.	Übersicht der Systemarchitektur von Aivaliotis u. a.	12
13.	Aufbau des AutoMate Frameworks	12
14.	6D-Posenbestimmung eines starren Objektes	13
15.	Unterteilung der 6D-Posenbestimmung nach Ebene und Methode	14
16.	Die Datensätze der BOP-Core Challenge	20
17.	Datensätze der neuen BOP-H3 Challenge mit Fokus auf Hand-Interaktion	21
18.	Die 21 Objekte des YCB-V Datensatzes	22
19.	Objekte und Szenen aus dem T-LESS Datensatz	23
20.	Aufbau zur Aufnahme der T-LESS Daten	24
21.	Exemplarische Aufnahme des ITODD Datensatzes	24
22.	Aufbau zur Datenaufnahme des ITODD Datensatzes	25
23.	Die Objekte des HOPE Datensatzes	25
24.	Unterschiedliche Umgebungen im HOPE Datensatz	26
25.	Unterschiedliche Objektanordnungen im HOPE Datensatz	26
26.	Entwicklung der Methoden zur 6D-Posenbestimmung	28
27.	Methoden zur 6D-Posenbestimmung nach bekannten und neuartigen Objekten	29
28.	Methoden zur 6D-Posenbestimmung nach RGB, RGB-D und reinen Tiefendaten	29
29.	Methoden zur 6D-Posenbestimmung nach Objektdatensätzen YCB-V und T-LESS	30
30.	Verwendete Methoden zur 6D-Posenbestimmung im Vergleich für die Datensätze YCB-V und T-LESS	31
31.	Ablauf der groben und feinen Posenbestimmung mit MegaPose	32
32.	Übersicht des Ablaufs von SAM-6D	34
33.	Beispiel für das Segmentieren und Erstellen von Masken mit dem SAM-Modell	35

34.	Übersicht des FoundationPose Frameworks	36
35.	Vergleich zwischen zufälliger Texturüberblendung und der von FoundationPose verwendeten Methode	37
36.	Übersicht der Architektur des HRC-Hubs	38
37.	Funktionsweise des Hand-Interaktionsmodells	39
38.	Unterschiedliche Methoden zur Posenbestimmung für das Tracking von Objekten	39
39.	Schematische Darstellung der Arbeitsziele	40
40.	Schematische Darstellung des Lochkamera-Modells	41
41.	Effekt der Verzeichnung am Beispiel eine Schachbrettmusters	43
42.	Positionsgenauigkeit und -Wiederholgenauigkeit nach DIN EN ISO 9283	44
43.	Orientierungsgenauigkeit und -Wiederholgenauigkeit für a nach DIN EN ISO 9283	46
44.	Genauigkeit als Differenz zwischen geschätzter Pose und Ground-Truth	47
45.	Schematisches Kameramodell zur Auflösung der Bildebene	48
46.	Relative Pixelgröße in Abhängigkeit der Auflösung	49
47.	Relative Pixelgröße in Abhängigkeit des FoV	49
48.	Relative Pixelgröße der Sensoren nach horizontaler Auflösung	50
49.	Exemplarische Darstellung der drei Roboterbewegungen zur Datenaufnahme	52
50.	Übersicht der verschiedenen Arten der Objektverdeckung durch Hand-Objekt- Interaktionen für die Versuchsreihen	52
51.	3D-gedruckter Halter zur Befestigung der Sensoren	55
52.	3-Punkt-Methode zum Kalibrieren eines Base-Koordinatensystems	55
53.	Kalibrierung des verwendeten KUKA LBR iiwa	56
54.	Beispiel für die ausgeführten Transformationen	58
55.	Beispiel für das plotten mehrerer Posen mit pytransform3d	59
56.	Veranschaulichung der Referenzierung eines Sensors über ArUco-Marker	60
57.	Beispiel für die ArUco Posenbestimmung	62
58.	Beispiele für notwendige Mindestanzahl an sichtbaren ArUco-Markern für die Po- senbestimmung eines ArUco-Gitters mit OpenCV	62
59.	Ausgabe des Programms zum Erstellen der Objekt-Referenzmarkierungen	63
60.	Darstellung eines Templates mit ArUco-Markern, Koordinatensystemen, Referenz- markierungen der Objekte und Beschriftung, sowie Bemaßung	64
61.	Aufgenommene Daten und mit BlenderProc erstelle Objektmaske	65
62.	Beispiel für RGB und Tiefenbild der RealSense D435i ohne Angleichung	67
63.	Schematische Darstellung der Zeitverzögerung der Sensorsignale	67
64.	Beispiele für Visualisierung der Posenbestimmung mit MegaPose	69
65.	Beispiel für Visualisierung der Posenbestimmung mit SAM6D	70
66.	Beispiel für Visualisierung der Posenbestimmung mit FoundationPose	71

67.	Beispiel für mögliche Fehler bei der 6D-Posenbestimmung	72
68.	Übersicht über den Gesamtprozess mit Teilschritten	73
69.	Erreichte Genauigkeit AP_P und RP_l der verschiedenen Methoden	75
70.	Erreichte Orientierungs- und Orientierungs-Wiederholgenauigkeit von MegaPose .	76
71.	Erreichte Orientierungs- und Orientierungs-Wiederholgenauigkeit von FoundationPose	77
72.	Beispiel für nicht eindeutige Ansichten der Objekte obj_000002 und obj_000014	77
73.	Erreichte Genauigkeit AP_P und RP_l nach Bewegung mit MegaPose	78
74.	Erreichte Genauigkeit AP_P und RP_l nach Bewegung mit FoundationPose	79
75.	Vergleich der erreichten Genauigkeiten AP_P und RP_l nach Sensor mit MegaPose	80
76.	Vergleich der erreichten Genauigkeiten AP_P und RP_l nach Sensor mit FoundationPose	80
77.	Erreichte Genauigkeit AP_P und RP_l nach Auflösung mit MegaPose	81
78.	Erreichte Genauigkeit AP_P und RP_l nach Auflösung mit FoundationPose	81
79.	Erreichte Genauigkeit AP_P und RP_l nach Verdeckung mit MegaPose	82
80.	Erreichte Genauigkeit AP_P und RP_l nach Verdeckung mit FoundationPose	83
81.	Genauigkeit während der circ_far Bewegung mit Handverdeckung mit MegaPose für obj_000021	83
82.	Genauigkeit während der circ_far Bewegung mit Handverdeckung mit FoundationPose für obj_000021	84
83.	Erreichte Genauigkeit AP_P und RP_l nach Bildfrequenz FPS mit FoundationPose	85
84.	Erreichte Genauigkeit AP_P und RP_l nach Robotergeschwindigkeit VEL mit FoundationPose	85
85.	Erreichte Genauigkeit AP_P und RP_l bei mehrfacher Wiederholung mit FoundationPose	86
86.	Ausblick auf zukünftige Forschung	88

Tabellenverzeichnis

1.	Vor- und Nachteile unterschiedlicher Sensoren für die Gestenerkennung in der MRK	9
2.	Vergleich von Intel RealSense D435i und Luxonis OAK-D Pro	27
3.	Vergleich von FoundPose und MegaPose auf den T-LESS und YCB-V Datensätzen	33
4.	Übersicht der verwendeten Testobjekte	51
5.	Übersicht der Versuchsreihen	53
6.	Übersicht der erwarteten CAD-Maßstäbe und Ausgabe-Formate der Methoden .	74
7.	Ergebnisse der Bewertungsmetrik für wiederholte Datenaufnahmen und Messungen	87

Quellcodeverzeichnis

1.	Implementierung zur Berechnung der Pose-Widerholgenauigkeit	47
2.	Implementierung zur Berechnung der Orientierungs-Widerholgenauigkeiten . . .	47
3.	Bewegungsbefehle des Roboters zur Datenaufnahme der Objekte	57
4.	Bestimmung der genauen Pose des Roboters über die Funktion getCurrentCartesianPosition	57
5.	Festlegung des verwendeten ArUco-Gitters mit OpenCV	61
6.	Erkennung und Posenbestimmung des ArUco-Gitters mit OpenCV	61
7.	Erstellen der Schnittebene der 3D-Modelle mit trimesh	64
8.	Beispiel für BlenderProc Ground-Truth Pose aus scene_gt.json	66
9.	Beispiel für BlenderProc Ground-Truth Bounding Box aus scene_gt_info.json . .	66
10.	Inhalt der camera_data.json für MegaPose	68
11.	Ausgabe der geschätzten Pose von MegaPose	69
12.	Inhalt der camera.json für SAM6D	70
13.	Anpassung der im PEM verwendeten Anzahl von Punkten	70
14.	Inhalt der cam_K.txt für FoundationPose	71

Abkürzungsverzeichnis

Abkürzung	Bezeichnung
ADD	engl.: Average Distance
ADD-S	engl.: Average Distance (with regards to) Symmetry
BOP	engl.: Benchmark for 6D Object Pose Estimation
CAD	engl.: Computer-Aided Design
FoV	engl.: Field of View
FPS	engl.: Frames Per Second
HMD	engl.: Head-Mounted Display
HOPE	engl.: Household Objects for Pose Estimation (Datensatz)
HRC	engl.: Human-Robot Collaboration
HRI	engl.: Human-Robot Interaction
ICP	engl.: Iterative Closest Point
IMU	engl.: Inertial Measurement Unit
ISM	engl.: Instance Segmentation Model (SAM-6D)
ITODD	engl.: Industrial 3D Object Detection Dataset (Datensatz)
KMU	Kleine und mittlere Unternehmen
MRK	Mensch-Roboter-Kollaboration
MSSD	engl.: Maximum Symmetry-Aware Surface Distance
MSPD	engl.: Maximum Symmetry-Aware Projection Distance
PbD	engl.: Programming by Demonstration
PdV	Programmieren durch Vormachen
PEM	engl.: Pose Estimation Model (SAM-6D)
PnP	engl.: Perspective-n-Point
RL	engl.: Reinforcement Learning
ROI	engl.: Region of Interest
ROS	engl.: Robot Operating System
T-LESS	engl.: Texture-Less (Datensatz)
VEL	engl.: Velocity
VSD	engl.: Visible Surface Discrepancy
YCB-V	engl.: Yale-CMU-Berkeley-Video (Datensatz)

1. Einleitung

An die moderne industrielle Produktion werden eine Vielzahl von Anforderungen gestellt [1]. Für produzierende Unternehmen, insbesondere solche mit einem hohen Anteil manueller Prozesse, ergibt sich hieraus ein komplexes Problem [2], dargestellt in Abbildung 1. Die fortschreitende Globalisierung führt zu einem verstärkten Preisdruck, welcher durch den intensiven internationalen Wettbewerb bedingt ist. Für Unternehmen, insbesondere in Hochlohnländern, erlangt die Wettbewerbsfähigkeit daher eine zunehmend hohe Relevanz. Gleichzeitig resultieren gesellschaftliche Entwicklungen sowie der demografische Wandel in einem zunehmenden Fachkräftemangel in diesen Ländern. Insbesondere in Verbindung mit steigenden Lohnkosten führt dies zu signifikanten ökonomischen Herausforderungen. Dies bedingt die Notwendigkeit einer zunehmenden Automatisierung. Diese muss jedoch ein hohes Maß an Flexibilität aufweisen, um auch anderen Anforderungen gerecht zu werden [3].

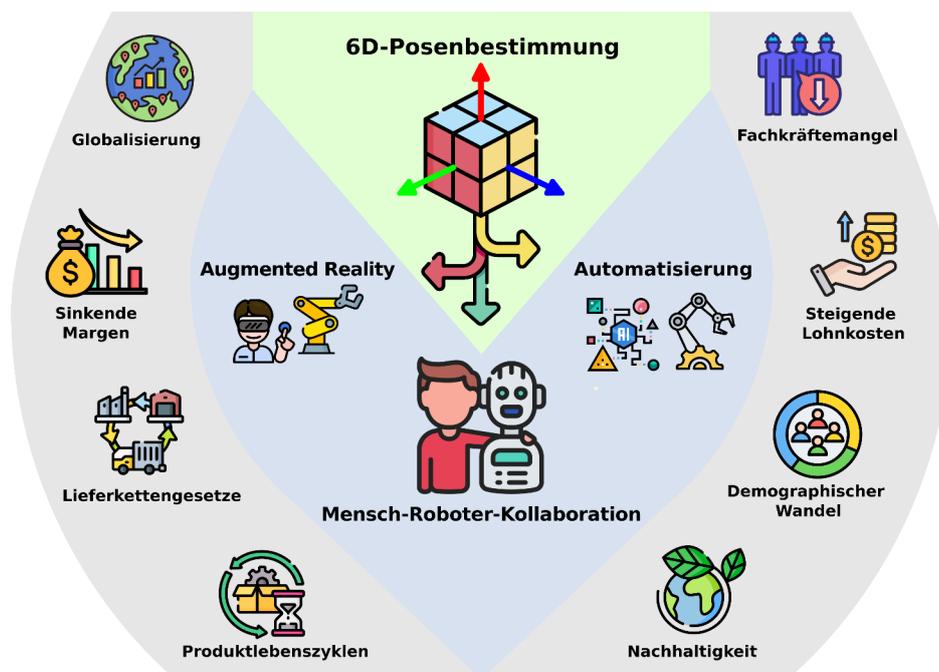


Abbildung 1: Kontext und Motivation der Arbeit

Darüber hinaus sind die Anforderungen an die Nachhaltigkeit der Produktion in den letzten Jahren deutlich gestiegen [4]. Dies ist zum einen auf gesetzliche Änderungen, zum anderen auf das Kundeninteresse zurückzuführen. Ein Beispiel hierfür ist das deutsche Lieferkettengesetz, welches die industrielle Produktion mit weitreichenden rechtlichen Verpflichtungen konfrontiert [5]. In Konsequenz kann dies den Bedarf an Automatisierung im Inland erhöhen, da die Auslagerung von Prozessen in Niedriglohnländer unter Umständen nicht mehr möglich ist. Seit Juli 2024 ist auch europaweit ein ähnliches Gesetz in Kraft getreten [6].

Die genannten Anforderungen sowie die daraus resultierenden Herausforderungen werden von Forschung und Industrie mit großem Engagement bearbeitet. Dies hat bereits zu einer Vielzahl von neuartigen Ansätzen und Verfahren geführt, welche das Potenzial besitzen, diese Herausforderungen zumindest teilweise zu lösen, siehe Abbildung 2.

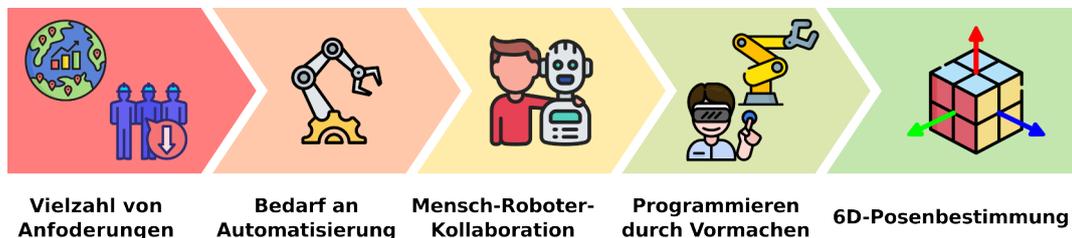


Abbildung 2: Herleitung des Ziels der Arbeit

Eine solche Entwicklung ist die **Mensch-Roboter-Kollaboration** (MRK, engl. Human-Robot-Collaboration HRC). Diese ermöglicht die Nutzung von Robotersystemen über den bisherigen Einsatz von Automatisierungssystemen hinaus, wodurch eine Steigerung der Gesamtproduktivität sowie eine Verbesserung der Produktqualität erzielt werden kann [7, 8]. Die MRK erfordert jedoch, wie die Automatisierung in der industriellen Produktion im Allgemeinen, eine umfangreiche Programmierung [9]. Diese Notwendigkeit ist von hoher Relevanz, da spezialisiertes Personal benötigt wird, welches entsprechende Kenntnisse und Fähigkeiten in der Programmierung und Wartung solcher Systeme besitzt. Der Mangel an qualifizierten Fachkräften in diesem Bereich vermindert somit das theoretische Potential, welches dieser Ansatz bietet [10].

Das Konzept **Programmieren durch Vormachen** (PdV) birgt wiederum das Potenzial, den Mangel an Fachpersonal im Bereich der Programmierung zu verringern. Es ermöglicht Anwendern ohne tiefgehende Programmierkenntnisse, Roboter und automatisierte Systeme zu programmieren [11]. Die Erstellung von Abläufen durch den Benutzer erfolgt durch einfaches Vormachen der entsprechenden Aufgaben, sodass der Roboter diese anschließend eigenständig wiederholt. In diesem Zusammenhang kann das PdV von weiteren neuen Entwicklungen, beispielsweise im Bereich der Augmented Reality (AR), unterstützt und erweitert werden. Der Erfolg dieses Ansatzes ist jedoch maßgeblich von der Verfügbarkeit robuster Methoden zur Detektion und Posensbestimmung von Objekten abhängig [12].

Gegenwärtig lassen sich beachtliche Erfolge und Verbesserungen bei der **6D-Posensbestimmung** im Vergleich zum Stand der letzten Jahre verzeichnen. Dies lässt sich zum einen darauf zurückführen, dass die neuen Methoden des maschinellen Lernens zum Einsatz kommen. Zum anderen ist die Verfügbarkeit von Datensätzen, die als Grundlage für das Training und die Evaluierung der Methoden dienen, in den letzten Jahren deutlich angestiegen [13, 14].

Ein wesentliches **Defizit** besteht jedoch darin, dass die derzeitige Evaluierung der 6D-Posenbestimmung lediglich dem Vergleich der Methoden untereinander dient. Qualitative Aussagen können getroffen werden, eine quantitative Aussage über die absolute Genauigkeit der Posenbestimmung ist jedoch nicht direkt möglich. Darüber hinaus ist die Anzahl der verfügbaren Datensätze noch nicht ausreichend, um eine Vielzahl von Anwendungsfällen adäquat bewerten zu können, da spezifische Objekteigenschaften nicht abgedeckt werden.

Das **Ziel dieser Arbeit** ist es daher, zu evaluieren, inwieweit aktuelle Methoden des Standes der Technik für den Einsatz in der intuitiven Programmierung durch Vormachen geeignet sind. Um dieses Ziel zu erreichen, soll ein automatisiertes Verfahren zur Aufnahme von annotierten (Test-)Daten entwickelt werden. Für die Annotation der Daten mit Ground-Truth soll ein Industrieroboter in Verbindung mit Templates eingesetzt werden, wodurch die Position und Orientierung der Objekte relativ zu Sensoren am Roboter bekannt ist. Anschließend sollen experimentelle Versuche mit aktuellen State-of-the-Art Methoden zur 6D-Posenbestimmung erste quantitative Aussagen über deren Genauigkeit ermöglichen. Dazu soll eine geeignete Metrik definiert und eingesetzt werden.

Die vorliegende Arbeit gliedert sich in mehrere Kapitel, die systematisch aufeinander aufbauen, siehe Abbildung 3. Zunächst werden in Kapitel 2 die theoretischen Grundlagen und der aktuelle Forschungsstand erläutert. Danach folgen in Kapitel 3 die Definition der spezifischen Ziele und die Forschungsfrage dieser Untersuchung. Anschließend wird in Kapitel 4 das wissenschaftliche Vorgehen sowie das konzeptionelle Design der Untersuchungen dargestellt. Das folgende Kapitel Kapitel 5 beschreibt die Umsetzung der entwickelten Konzepte in die Praxis sowie die Durchführung der Experimente bzw. Fallstudien. Die erzielten Ergebnisse werden im Kapitel Kapitel 6 dargestellt und kritisch analysiert. Abschließend gibt das Kapitel 7 einen Überblick über die wichtigsten Erkenntnisse und einen Ausblick auf mögliche zukünftige Forschungsarbeiten und Weiterentwicklungen im Bereich der 6D-Posenbestimmung für die MRK.



Abbildung 3: Gliederung der Arbeit

2. Grundlagen und Stand der Technik

In diesem Kapitel wird zunächst ein Überblick über die grundlegenden Konzepte der Mensch-Roboter-Kollaboration gegeben. Ein besonderer Fokus liegt dabei auf der Methode des Programmierens durch Vormachen, welche eine intuitive und benutzerfreundliche Art der Robotik-Programmierung darstellt. Im Anschluss daran werden die grundlegenden Aspekte der 6D-Posenbestimmung behandelt. Hierbei wird auf wesentliche Metriken, relevante Datensätze sowie aktuelle State-of-the-Art Methoden eingegangen, um die neuesten Entwicklungen und Herausforderungen auf diesem Gebiet zu beleuchten. Abschließend wird der sogenannte *HRC-Hub* vorgestellt, ein im Rahmen dieser Arbeit mitentwickeltes System, das als zentrale Plattform für die Erforschung und Implementierung von MRK-Technologien dient und den Kontext dieser Arbeit bildet.

2.1. Mensch-Roboter-Kollaboration

Die Mensch-Roboter-Kollaboration markiert einen entscheidenden Fortschritt in der industriellen Produktion, da sie die Stärken von Mensch und Maschine in einer Weise verbindet, die zuvor nicht möglich war. Die enge Zusammenarbeit von menschlichen Arbeitskräften und Robotern erlaubt eine flexible Automatisierung, wodurch die Vielseitigkeit und Anpassungsfähigkeit herkömmlicher Produktionsprozesse signifikant erhöht wird. Dies ist insbesondere dadurch möglich, dass die Präzision und Effizienz von Robotern mit der Kreativität und Problemlösungsfähigkeit von Menschen verbunden werden [7].

Diese innovative Technologie kann einen wesentlichen Beitrag zur Bewältigung aktueller Herausforderungen in der industriellen Produktion leisten (vgl. Kapitel 1). Die Entwicklung basiert auf langjährigen Forschungsarbeiten im Bereich der Robotik und ermöglicht die Zusammenarbeit von Mensch und Roboter in einem gemeinsamen Arbeitsraum [15]. Die Stufen der gemeinsamen Nutzung des Arbeitsraums von Mensch und Roboter umfassen Koexistenz, Kollaboration und Kooperation, siehe Abbildung 4. Eine weitere Möglichkeit der Unterteilung der Mensch-Roboter-Kollaboration besteht in einer Differenzierung unabhängig vom Arbeitsraum, wobei eine Klassifikation der Kooperationstypen vorgenommen wird. In diesem Kontext kann der Roboter zum einen als dritte Hand des Mitarbeiters dienen, zum anderen die Bewegungsvorgabe des Mitarbeiters imitieren (vgl. Abschnitt 2.2). Eine weitere Möglichkeit ist die koexistierende Interaktion zwischen Roboter und Mensch, wobei der Roboter dem Mitarbeiter beispielsweise Werkzeug anreicht [7].

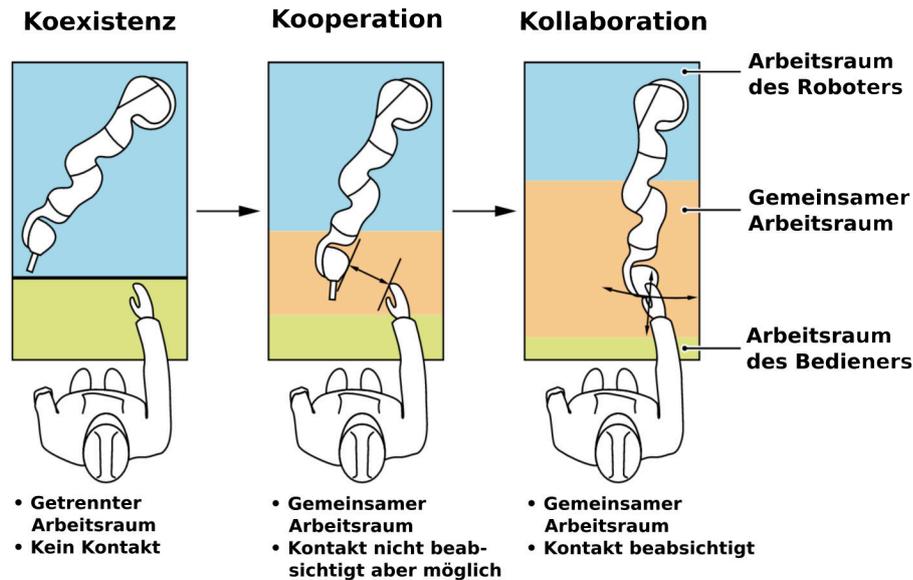


Abbildung 4: Entwicklung des Arbeitsraumes von sicherheits- zu kollaborationsorientiert, adaptiert aus [15]

Bei der Implementierung eines Fertigungsprozesses mit einer Mensch-Roboter-Kollaboration in der Produktion sind ähnliche Aspekte von Relevanz wie bei der traditionellen Automatisierung. In diesem Kontext sind insbesondere die *Wirtschaftlichkeit* der Lösung unter Berücksichtigung der Rentabilitätsanforderungen und Amortisationszeiten, die *Zuverlässigkeit*, welche eine hohe Anlagenverfügbarkeit und Fehlertoleranz umfasst, die Sicherstellung einer hohen und konstanten *Produktqualität*, der *Arbeits- und Gesundheitsschutz* sowie die *Flexibilität und Anpassungsfähigkeit des Systems* von entscheidender Bedeutung [7].

Um die Sicherheit von Menschen bei der Bedienung von MRK-Systemen zu gewährleisten, müssen Richtlinien und Standards eingehalten werden. Die Norm *DIN EN ISO 10218-1* definiert verschiedene Schutzmechanismen für eine sichere Mensch-Roboter-Kollaboration [16]:

- *Handgeführte Steuerungen* (engl.: Hand-guided controls): Die Bewegungen des Roboters werden aktiv vom Arbeiter kontrolliert, sodass der Arbeiter die Kontrolle über die Handlung hat.
- *Geschwindigkeits- und Abstandsüberwachung* (engl.: Speed and Separation Monitoring): Roboter verwenden Sensoren, um ihre Umgebung zu überwachen und verhindern so den Kontakt, wenn ein Mensch zu nah kommt.
- *Energie- und Kraftbegrenzung* (engl.: Power and Force Limiting): Durch inhärente Konstruktion oder Sicherheitsfunktionen wird gewährleistet, dass es zu keiner Verletzung des Mitarbeiters kommen kann. Sobald die Kontaktkräfte zwischen Mitarbeitern und Maschinen Grenzwerte überschreiten, geht der Roboter in einen Sicherheitshalt.

Die Umsetzung von MRK-Systemen erfordert den Einsatz geeigneter Sensortechnologie. Wie in traditionellen Produktionssystemen werden Sensoren zur Automatisierung von Fertigungsschritten verwendet, beispielsweise für Abstandsmessungen oder Anwesenheitskontrollen von Bauteilen. Im Kontext der MRK kommt ihnen jedoch auch eine entscheidende Rolle zur Gewährleistung der Systemsicherheit zu [7]. Im Gegensatz zur traditionellen Automatisierung ist die Mensch-Roboter-Kollaboration auch in besonderem Maße auf effiziente Schnittstellen zwischen Mensch und Maschine angewiesen, da die Interaktion zwischen beiden Akteuren eine wesentliche Rolle spielt. Diese Schnittstellen dienen nicht nur der Programmierung und Überwachung, sondern sind integraler Bestandteil der Arbeitsprozesse, in denen Menschen und Roboter in enger Wechselwirkung miteinander agieren. Die wissenschaftliche Disziplin, die sich mit diesem Gebiet befasst, wird als *Mensch-Roboter-Interaktion* (engl.: Human-Roboter-Interaction, HRI) bezeichnet. Die Interaktion zwischen Mensch und Roboter kann dabei auf verschiedenen Modalitäten und Techniken basieren, wie in Abbildung 5 exemplarisch dargestellt.



Abbildung 5: Modalitäten und Techniken der Mensch-Roboter-Interaktion, adaptiert aus [17]

In diesem Kontext weist *Augmented Reality* ein signifikantes Potenzial für die Mensch-Roboter-Kollaboration auf. AR ermöglicht eine Vielzahl an Interaktionsmöglichkeiten und stellt dem Menschen einen direkten visuellen Output bereit. Komplexe Arbeitsprozesse können durch den Einsatz von AR intuitiv visualisiert und gesteuert werden, was die Effizienz und Sicherheit in der MRK erheblich steigert. Diesbezüglich existiert bereits eine Vielzahl an vorgestellten Lösungen aus Forschung und Wirtschaft, welche das Potenzial von Augmented Reality aufzeigen [18, 19, 17]. Insbesondere die Verwendung von sogenannten *Head-Mounted-Displays* (HMD) hat sich hierbei als besonders vorteilhaft erwiesen, da es dem Bediener ermöglicht, direkt auf die visuelle Führung vor seinem Blickfeld zuzugreifen, während er beide Hände frei hat [20]. Die visuelle Augmentation kann mit unterschiedlichen Zielsetzungen eingesetzt werden. Dies wird in Abbildung 6 veranschaulicht.



Abbildung 6: Ziele und Vorteile der visuellen Augmentation, adaptiert aus [17]

2.2. Programmieren durch Vormachen

Im vorherigen Kapitel erfolgte eine erste Auseinandersetzung mit verschiedenen Ansätzen und Technologien der Mensch-Roboter-Kollaboration. Ein besonders vielversprechender Bereich innerhalb der Mensch-Roboter-Kollaboration stellt das sogenannte *Programmieren durch Vormachen* (engl.: Programming by Demonstration, PbD) dar. Diese Methode erlaubt es den Anwendern, Roboter durch eine einfache Demonstration von Aufgaben zu programmieren, wodurch eine erhebliche Reduktion der komplexen und zeitaufwendigen manuellen Programmierarbeit erzielt wird [21]. Das Programmieren durch Vormachen zählt zu den benutzerzentrierten Methoden der Roboterprogrammierung, siehe Abbildung 7.

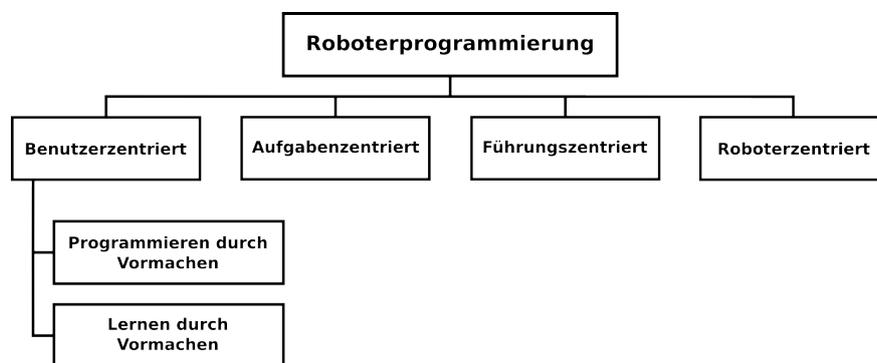


Abbildung 7: Einteilung der Roboterprogrammierung für die MRK in vier Ansätze und Unterscheidung der benutzerzentrierten-Programmierung, adaptiert aus [7]

Beim Programmieren durch Vormachen demonstriert der Benutzer eine beispielhafte Lösung einer Aufgabe dem Robotersystem. Dabei zeigt der Benutzer typischerweise alle erforderlichen Teilschritte. Der Roboter entwickelt daraus eine Strategie, die auf Sensoren basiert, um das vorgeführte Verhalten nachzubilden [7]. Hierbei lassen sich grundsätzlich zwei Konzepte unterscheiden, nämlich das *Programmieren durch Demonstration* und das *Programmieren durch Imitation*, vgl. Abbildung 8.

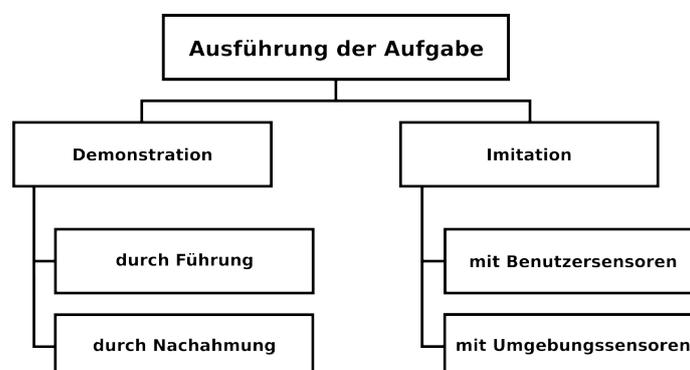


Abbildung 8: Strukturierung von Programmieren durch Vormachen, adaptiert aus [7] nach [22]

Die *Programmierung durch Demonstration* bezeichnet ein Vorgehen, bei dem die Aufgabenstellung direkt am Roboter oder an einer identischen Plattform demonstriert und von den Sensoren des Robotersystems erfasst wird, sodass eine exakte Beobachtung der Ausführung des Roboters gewährleistet ist. Es lassen sich zwei untergeordnete Ansätze unterscheiden [7]:

- Bei der *Demonstration durch Führung* erfolgt die Interaktion zwischen Benutzer und Roboter unmittelbar (Teleoperation), sodass eine präzise Wahrnehmung der Aufgabe gewährleistet ist. Dies bedingt einen unmittelbaren Informationsaustausch, vorausgesetzt, der Roboter lässt sich einfach steuern und ist mit adäquaten Sensoren ausgestattet.
- Bei der *Demonstration durch Nachahmung* ahmt der Roboter die Ausführung des Benutzers nach. Die beobachtete Ausführung ist nicht exakt die gleiche wie die des Benutzers, sondern der Roboter versucht, die Aufgabe anhand der Demonstration zu reproduzieren. Hierfür sind zusätzliche Komponenten zur Verfolgung und Nachahmung erforderlich.

Beim *Programmieren durch Imitation* wird die zu erledigende Aufgabe zunächst vom Benutzer vorgeführt und anschließend lediglich über Sensoren beobachtet. Dies impliziert, dass die Ausführung nicht direkt vom Roboter übernommen werden kann, sondern in roboterbezogene Anweisungen übersetzt werden muss. In Bezug auf die Sensorik lassen sich erneut zwei untergeordnete Ansätze unterscheiden [7]:

- Bei der *Imitation mit Benutzersensoren* sind die Sensoren am Benutzer angebracht, was eine genaue und natürliche Durchführung der Aufgabe ermöglicht. Allerdings können die zusätzliche Sensorik und Verkabelung die Bewegungsfreiheit des Benutzers einschränken.
- Bei der *Imitation mit Umgebungssensoren* sind die Sensoren in der Umgebung oder am Roboter angebracht. Hier müssen die notwendigen Bewegungen aus der Beobachtung extrahiert werden, was höhere Anforderungen an die Sensorik und deren Auswertung stellt.

Es existiert eine Vielzahl unterschiedlicher Sensortechnologien, die grundsätzlich für Gestenerkennung beim PdV verwendet werden können, siehe Abbildung 9.

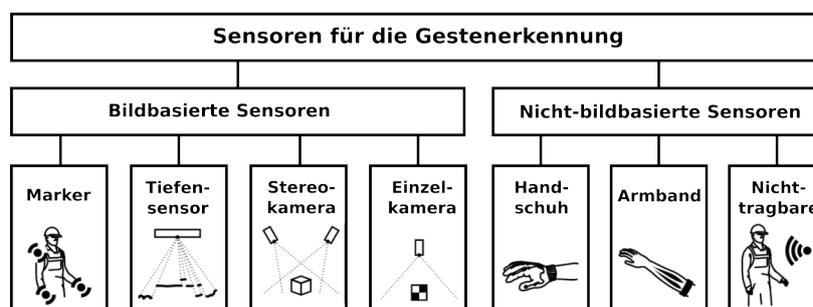


Abbildung 9: Verschiedene Arten von Sensoren für die Gestenerkennung, adaptiert aus [23]

Diese Sensoren lassen sich in zwei grundlegende Kategorien der Datenerfassung unterteilen: bildbasierte und nicht-bildbasierte Ansätze. Die folgende Tabelle 1 gibt einen vergleichenden Überblick über die verschiedenen Sensortechnologien zur Gestenerkennung im Rahmen der Programmierung durch Vormachen. Darüber hinaus werden die Vor- und Nachteile der verschiedenen Ansätze dargestellt.

Tabelle 1: Vor- und Nachteile unterschiedlicher Sensoren für die Gestenerkennung in der MRK, adaptiert aus [23]

Sensor	Vorteile	Nachteile
Marker	Geringer Aufwand für die Berechnung	Marker auf dem Körper des Nutzers
Einzelkamera	Einfache Einrichtung	Vergleichsweise störanfällig
TOF-Kamera	Sehr hohe Bildrate	Hohe rechnerische Komplexität, Kalibrierung notwendig
Strukturierter Lichtsensor	Schnelle Entwicklung der Software-Unterstützung für die Erkennung von Körpergesten	Kann nicht für die Erkennung von Handgesten über 2 m verwendet werden
Handschuh	Schnelle Reaktion, präzise Verfolgung	Unhandliches Gerät durch externe Kabel
Armband	Schnelle Reaktion, großer Erfassungsbereich	Muss Kontakt zum Nutzer haben
Nicht-tragbare Sensoren	Vermeidet Kontakt mit Körper des Nutzers	Geringe Auflösung, Technologie noch nicht ausgereift

Die bisherigen Forschungsergebnisse legen die Schlussfolgerung nahe, dass keine Technologie für sämtliche Anwendungsbereiche gleichermaßen geeignet ist. Für den Einsatz im Innenbereich erweist sich die Verwendung von Tiefensensoren als vielversprechendste bildbasierte Technologie. Der Vorteil dieser Technologie liegt in der einfachen Einrichtung, Kalibrierung sowie der unkomplizierten Datenverarbeitung. Des Weiteren besteht eine große Entwicklergemeinschaft, die eine Vielzahl potenzieller Lösungen bereitstellen kann [23]. In diesem Kontext können auch ab Werk kalibrierte Stereo-Kameras berücksichtigt werden, wie sie in Abschnitt 2.4.3 vorgestellt werden. Diese Modelle bieten die gleichen Vorteile und können darüber hinaus als Einzelkamera fungieren, wodurch ein erweitertes Spektrum an Methoden anwendbar ist.

2.3. Beispiele aus der aktuellen Forschung

Im Folgenden werden ausgewählte aktuelle Forschungsbeispiele präsentiert, welche den zuvor vorgestellten Ansatz des Programmierens durch Vormachen sowie darauf aufbauender Methoden verwenden, um die Programmierung von Robotern zu vereinfachen.

Zhang, Wang und Xiong präsentieren ein Framework für die Programmierung von Industrierobotern. Dieses Framework umfasst die Identifizierung und Segmentierung von Aktionen, die Formulierung von Abläufen sowie die Abbildung und Optimierung von Roboterbewegungen. Die Ergebnisse der finalen Experimente mit Klemmbausteinen in RobotStudio, einer von ABB entwickelten Simulationsumgebung für Industrieroboter, demonstrieren die Realisierbarkeit der vorgeschlagenen PdV-Lösung [24]. Des Weiteren nutzen die Autoren das CAD-Modell einer Baugruppe, um daraus Montageeinschränkungen abzuleiten. Die Posenbestimmung der Objekte erfolgt im Rahmen der Arbeit ohne die Verwendung von zusätzlichen Markern. Dies ist möglich, da es sich mit Klemmbausteinen um relativ einfache Objekte handelt und diese unter idealen Bedingungen, nämlich vereinzelt und auf komplett schwarzem Untergrund, detektiert werden.

Einen ähnlichen Ansatz verfolgen Zou und Andulkar. Die Methode generiert zunächst eine Montagereihenfolge auf Basis zweier unterschiedlicher Vorgaben. Im ersten Schritt werden Montagebeziehungen aus CAD-Modellen zwischen den zu montierenden Teilen extrahiert. Des Weiteren erfolgt eine Demonstration der Montageaufgabe durch den Nutzer an virtuellen Objekten¹. Auf Basis dieser Daten wird zunächst eine Montagereihenfolge abgeleitet, welche unter Verwendung einer Bibliothek von Montagevorgängen zu einer kompletten Aktionssequenz der Montage formuliert wird. Die Aktionssequenz wird zunächst in einer Simulation ausgeführt und validiert. Im Anschluss an die erfolgreiche Durchführung erfolgt die Ausführung sowie die experimentelle Validierung auf einem realen Roboter, siehe Abbildung 10. Die Ergebnisse zeigen, dass die Methode intuitiv ist und der Roboter von ungelerten Bedienern einfach programmiert werden kann [25].

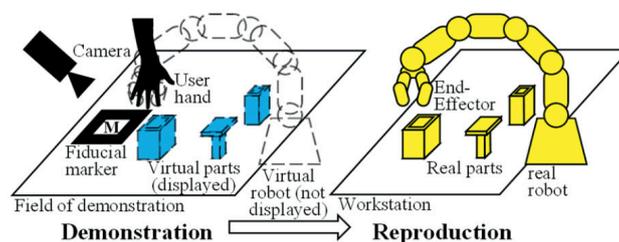
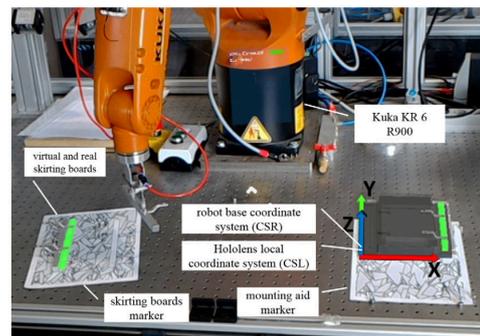
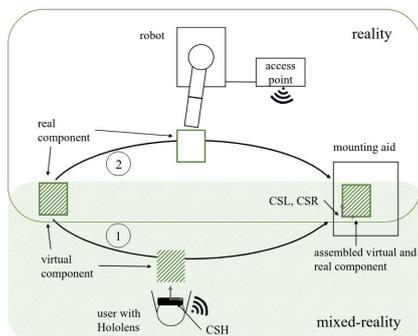


Abbildung 10: Übersicht der Methode von Zou und Andulkar [25]

¹Der vorliegenden Quelle lässt sich nicht eindeutig entnehmen, auf welche Art und Weise die virtuellen Objekte dem Nutzer präsentiert werden.

Blankemeyer u. a. präsentieren einen innovativen Ansatz zur vereinfachten Programmierung von Robotern mittels AR. Das Hauptziel dieses Ansatzes ist es, den Programmieraufwand zu minimieren und die Flexibilität in der industriellen Produktion zu erhöhen, indem die benötigten Programmierkenntnisse der Mitarbeiter reduziert werden. Kern des Ansatzes ist die Nutzung der Microsoft HoloLens. Diese ermöglicht das Erkennen realer Komponenten durch optische Marker und deren Überlagerung mit virtuellen CAD-Modellen. Der Prozess der virtuellen Montage und Programmierung gestaltet sich wie folgt: Der Bediener bewegt die virtuellen Komponenten an die gewünschten Positionen, wobei diese Bewegungen von der HoloLens erfasst werden. Die erfassten Start- und Endkoordinaten werden im Koordinatensystem des Roboters gespeichert und transformiert, sodass der Roboter die Montageschritte in der Realität nachahmen kann. Die eigentliche Programmierlogik wird auf dem Robotercontroller implementiert, der die Pfadplanung zwischen den einzelnen Positionen übernimmt. Als Beispielanwendung zeigen die Autoren die Montage eines Helikoptermodells. Hierbei werden zwei Modellkomponenten virtuell zusammengebaut, und die resultierenden Bewegungsprofile an den Roboter übertragen, der diese Echtzeit-Montage dann ausführt. Die Autoren validieren Ihren Ansatz in praktischen Versuchen mit einem Kuka KR6 R900 sixx. Die Datenübertragung erfolgt dabei mit dem Kuka Robot-Sensor-Interface [26], siehe Abbildung 11.



(a) Schematischer Aufbau und Ablauf der virtuellen Programmierung (b) Montagezelle mit Roboter, Markern, realen und virtuellen Objekten

Abbildung 11: Verfahren und Versuchsaufbau der Methode von Blankemeyer u. a. [26]

Aivaliotis u. a. stellen in Ihrer Arbeit eine Augmented Reality Software-Suite vor, die die Interaktion der Bediener mit flexiblen mobilen Roboterarbeitern unterstützt. Die entwickelte AR-Suite, integriert mit dem Digital Twin der Produktionsstätte, bietet den Bedienern eine virtuelle Schnittstellen zur einfachen Programmierung der mobilen Roboter. Darüber hinaus stellt sie Informationen über den Prozess und den Produktionsstatus dar und schafft ein besseres Sicherheitsbewusstsein durch Überlagerung der aktiven Sicherheitszonen um die Roboter. Im Falle unerwarteter Ereignisse gibt sie dem Nutzer Anweisungen. Die vorgeschlagene Software wurde für das Microsoft HoloLens Mixed Reality Headset 2 entwickelt, integriert mit dem Robotic Operating System (ROS), und wurde in einer Fallstudie aus der Automobilindustrie getestet [27], vgl. Abbildung 12.

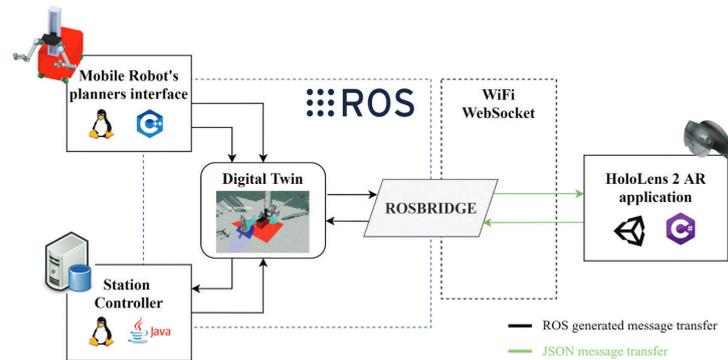


Abbildung 12: Übersicht der Systemarchitektur von Aivaliotis u. a. [27]

Aufbauend auf der PdV-Idee gibt es in jüngster Zeit Forschungsprojekte, die aktuelle Methoden des maschinellen Lernens nutzen, um den Zeit- und Arbeitsaufwand weiter zu reduzieren. Mandlekar u. a. stellen einen *Human-in-the-Loop Task and Motion Planning* Ansatz vor. Dabei handelt es sich um ein neuartiges System, das die Vorteile von PdV und *Task and Motion Planning (TAMP)* kombinieren soll. Dazu verwendet das System Kontrollmechanismen, die selektiv die Kontrolle an einen menschlichen Teleoperator übergeben und wieder entziehen, was eine effiziente Datenerfassung ermöglicht. Die gesammelten menschlichen Daten werden mit einem Imitationslernrahmen (vgl. Abschnitt 2.2) kombiniert, um eine TAMP-gesteuerte Politik zu trainieren. Dies führt zu einer besseren Leistung im Vergleich zum Training mit vollständiger PdV [28]. Die Arbeit von Tang u. a. präsentiert ein System, welches in der Lage ist, unterschiedliche Montageaufgaben ohne menschliche Intervention zu erlernen. Zu diesem Zweck wird eine neuartige Kombination von drei unterschiedlichen Algorithmen verwendet. Um ohne menschliche Demonstration auszukommen, leitet das System Montageabläufe aus automatisierten Demontagevorgängen ab, indem diese umkehrt werden. Aufbauend auf den ermittelten Demontageabläufen werden durch *Reinforcement Learning* (RL) Roboter-Trajektorien abgeleitet, welche den Ablauf imitieren sollen. Als Belohnungsfunktion für das Reinforcement Learning wird das *Dynamic Time Warping* eingesetzt, ein Algorithmus zur Messung der Ähnlichkeit zweier zeitabhängiger Datenreihen. Die Erfolgsrate des Verfahrens liegt bei 80% bei 20 Montageaufgaben [29]. Beiden Verfahren basieren in erheblichem Maße auf Simulationsmodellen und wurden darüber hinaus an realen Industrierobotern evaluiert.

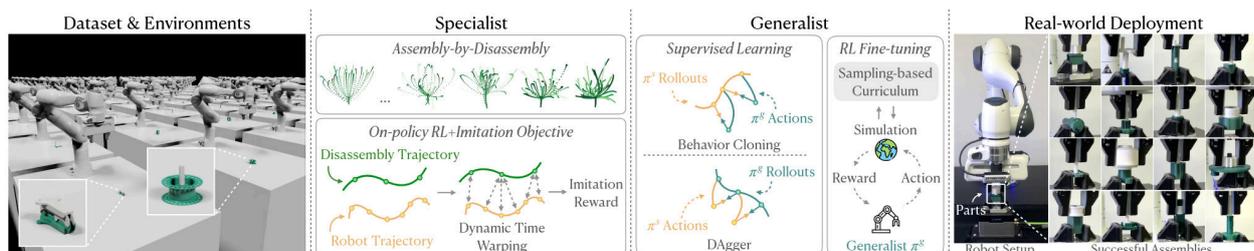


Abbildung 13: Aufbau des AutoMate Frameworks [29]

2.4. Grundlagen der 6D-Posenbestimmung

Die exakte Bestimmung der Position und Orientierung eines Objekts (6D-Posenbestimmung) spielt in zahlreichen technologischen und wissenschaftlichen Anwendungen eine entscheidende Rolle. Dies umfasst Bereiche wie Robotik und Automation, Augmented Reality sowie medizinische Bildgebung. Die Fähigkeit, die Pose eines Objekts im Raum präzise und zuverlässig zu bestimmen, eröffnet vielfältige Möglichkeiten und verbessert die Effizienz und Genauigkeit der jeweiligen Systeme erheblich [30, 13]. Dieses Kapitel widmet sich den grundlegenden Konzepten, Methoden und aktuellen Forschungsergebnissen im Bereich der 6D-Posenbestimmung und zeigt auf, wie diese in verschiedenen Bereichen eingesetzt und weiterentwickelt werden können.

In der vorliegenden Arbeit wird die Problematik der **6D-Posenbestimmung spezifischer, starrer Objekte** unter Zuhilfenahme verfügbarer 3D-Modelle (CAD-Modelle) behandelt. Das zentrale Problem besteht darin, bei gegebenen 3D-Modellen, die 6D-Posen von möglicherweise mehreren Instanzen von möglicherweise mehreren Objektklassen aus einem einzigen Testbild zu schätzen. Testbilder können hierbei RGB, RGB-D, oder ausschließlich Tiefenkanäle besitzen. Zudem werden die intrinsischen Kameraparameter als bekannt vorausgesetzt (vgl. Abschnitt 4.1). Wichtig ist anzumerken, dass die Region of Interest (ROI) den Methoden für die Posenbestimmung bereitgestellt wird. Die ROI umfasst die Region des Bildes, in der sich das Objekt befindet. Die Objekterkennung, insbesondere von unbekanntem Objekten, ist ein separates Problem und wird in diesem Zusammenhang nicht näher behandelt (vgl. [14]).

Die Größe und Form eines starren Objekts sind festgelegt und bleiben unverändert, sofern keine Kräfte auf das Objekt einwirken. Die Position eines starren Objekts wird folglich vollständig durch eine starre Transformation mit sechs Freiheitsgraden (6D-Pose) beschrieben. Diese Pose besteht aus drei Translationskomponenten (x, y, z) sowie drei Rotationskomponenten ($roll, pitch, yaw$) und definiert die Transformation vom 3D-Koordinatensystem des Objektmodells (KS_{Objekt}) zum 3D-Koordinatensystem der Kamera (KS_{Kamera}) [30], siehe Abbildung 14.

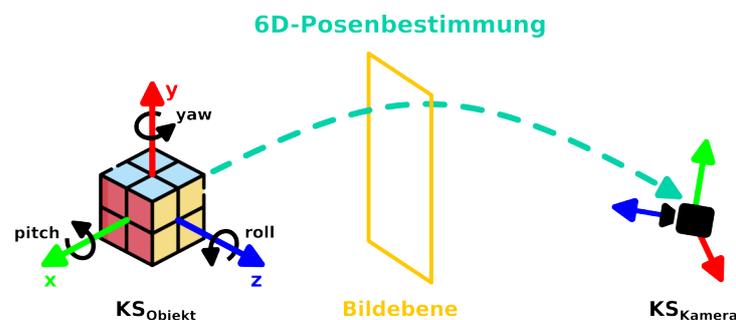


Abbildung 14: 6D-Posenbestimmung eines starren Objektes, in Anlehnung an [30]

Die **Posenbestimmung neuartiger Objekte** (engl.: novel objects) zeichnet sich durch die Fähigkeit aus, die Pose von Objekten zu bestimmen, die zuvor weder in spezifischen Trainingsdatensätzen verwendet wurden noch bekannt waren. Diese Methode nutzt fortschrittliche Algorithmen und maschinelles Lernen, um Objekte in einer Szene zu erkennen und ihre Pose zu schätzen, ohne auf explizite, vorab bekannte Modelle der jeweiligen Objektinstanzen angewiesen zu sein. Eine der größten Stärken dieser Posenbestimmung ist ihre Flexibilität: Da sie nicht von vorgefertigten Trainingsdaten abhängt, kann sie in einer Vielzahl von Szenarien und für unterschiedliche Anwendungsfälle eingesetzt werden [14]. Dieser Vorteil ist im Kontext des größeren Zusammenhangs und der Ausgangssituation dieser Arbeit besonders relevant (vgl. Kapitel 1). Die Methoden, welche im weiteren Verlauf dieser Arbeit untersucht werden, beschränken sich deswegen auf solche, die die Posenbestimmung von neuartigen Objekte ermöglichen.

Eine Klassifikation der 6D-Posenbestimmung kann auf unterschiedliche Weise erfolgen. Eine mögliche Einteilung nach Ebene und Methode wird von Guan u. a. in Abbildung 15 vorgestellt.

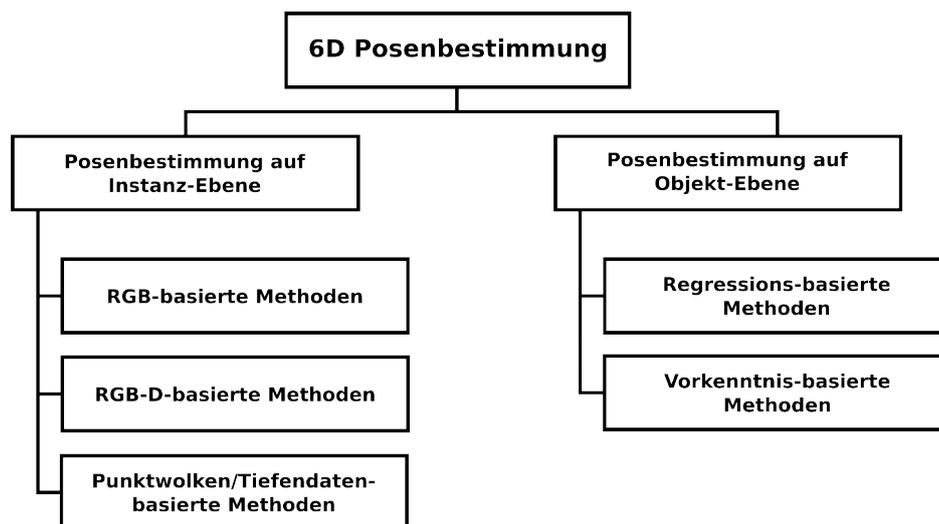


Abbildung 15: Unterteilung der 6D-Posenbestimmung nach Ebene und Methode, adaptiert aus [13]

Im Rahmen der **Posenbestimmung auf Objekt-Ebene** sind die CAD-Modelle der Testobjekte nicht bekannt, jedoch wird angenommen, dass die Objekte einer bekannten Kategorie angehören. Die genannten Methoden basieren auf gemeinsamen Eigenschaften von Objekten innerhalb der jeweiligen Kategorien, sodass eine Verallgemeinerung auf unbekannte Kategorien nicht möglich ist. Für eine Vielzahl von Objekten, insbesondere solche aus dem industriellen Kontext, lassen sich die jeweiligen Kategorien nicht klar voneinander abgrenzen [31]. Dies führt dazu, dass die genannten Methoden in diesen Anwendungsfällen keine geeignete Lösung darstellen. Im weiteren Verlauf dieser Arbeit wird diese Art von Posenbestimmung daher nicht weiter betrachtet.

Demgegenüber steht die **Posenbestimmung auf Instanz-Ebene**. Das Ziel dieser Bestimmung ist die exakte Festlegung der Pose einer spezifischen Objektinstanz. Dabei entspricht das zu erkennende Objekt exakt dem bereitgestellten 3D-Modell. Die Posenbestimmung auf Instanz-Ebene lässt sich in drei Hauptkategorien einteilen (vgl. [13]):

- **RGB-basierte Methoden:** Die Einschätzung der Pose erfolgt ausschließlich unter Zuhilfenahme von Farbbildern, wobei typischerweise Convolutional Neural Networks zum Einsatz kommen. Allerdings zeigen sich diese Methoden verhältnismäßig anfällig gegenüber schlechten Lichtverhältnissen und Verdeckungen.
- **RGB-D-basierte Methoden:** Die Kombination von RGB-Bildern mit Tiefeninformationen, welche durch Sensoren wie Kinect oder RealSense gewonnen werden, ermöglicht eine robustere und exaktere Ergebnisermittlung. Die zusätzliche Berücksichtigung der Geometrieinformation führt zu einer Verbesserung der Genauigkeit und Zuverlässigkeit der Ergebnisse.
- **Punktwolken/Tiefendaten-basierte Methoden:** Verwenden ausschließlich Tiefendaten oder Punktwolken von 3D-Scannern oder Lidar-Sensoren. Algorithmen wie Iterative Closest Point (ICP) erlauben eine präzise Bestimmung der Pose anhand der geometrischen Struktur der Objekte. Dies gewährleistet eine hohe Robustheit gegenüber Lichtverhältnissen und Beleuchtungsänderungen. Allerdings besteht dadurch bedingt auch eine starke Limitation der Objekte, deren Posenbestimmung möglich ist. So ist es beispielsweise nicht möglich, die Orientierung einer symmetrischen Geometrie zu bestimmen, die sich lediglich durch ihre Textur unterscheidet.

Im Folgenden werden zunächst die Bewertungsmetriken vorgestellt, die in der aktuellen Forschung zur Bewertung der Güte einer 6D-Posenbestimmung verwendet werden. Anschließend werden zur besseren Verständlichkeit einige der Datensätze erläutert, die für diese Bewertungen verwendet werden. Nach einer kurzen Auseinandersetzung mit den Sensoren für die 6D-Posenbestimmung schließt dieses Unterkapitel mit der Vorstellung aktueller Methoden zur 6D-Posenbestimmung.

2.4.1. Bewertungsmetriken für die 6D-Posenbestimmung

In der Literatur werden verschiedene Metriken zur Bewertung von Methoden für die 6D-Posenbestimmung diskutiert. Diese Metriken bewerten die Qualität einer Posenbestimmung, indem sie die Übereinstimmung zwischen der bestimmten Pose und der tatsächlichen Pose des Objekts angeben. Die tatsächliche Pose wird in diesem Kontext als Ground-Truth bezeichnet.

Im Folgenden sollen einige relevante Bewertungsmetriken kurz vorgestellt werden. Hierbei soll verdeutlicht werden, dass die aktuell verwendeten Metriken zwar einen guten Vergleich der Methoden untereinander ermöglichen, jedoch keine absoluten und expliziten Aussagen über die erreichbare Genauigkeit erlauben.

Eine der ersten wissenschaftlichen Arbeiten, die sich explizit mit der Bewertung von 6D-Posenbestimmungen auseinandersetzt, ist in *On Evaluation of 6D Object Pose Estimation* zu finden [32]. Hodaň, Matas und Obdržálek stellen fest, dass die Bewertung von 6D-Objektposenschätzungen nicht trivial ist. Da aufgrund von Objekt-Symmetrien und Verdeckungen die Objektpose mehrdeutig sein kann, das heißt, es kann mehrere Objektposen geben, die in dem gegebenen Bild nicht zu unterscheiden sind und daher als gleichwertig behandelt werden sollten.

Also Modellunabhängige Fehlerfunktionen führen Hodaň, Matas und Obdržálek zunächst den translatorischen Fehler e_{TE} auf ([32] nach [33]):

$$e_{TE}(\hat{t}, \bar{t}) = \|\bar{t} - \hat{t}\|^2 \quad (2.1)$$

Wobei $\hat{P} = (\hat{R}, \hat{t})$ die geschätzte Pose ist und $\bar{P} = (\bar{R}, \bar{t})$ der Ground-Truth entspricht. In analoger Weise lässt sich der rotatorische Fehler e_{RE} bestimmen ([32] nach [34]):

$$e_{RE}(\hat{R}, \bar{R}) = \arccos\left(\frac{\text{Spur}(\hat{R}\bar{R}^{-1}) - 1}{2}\right) \quad (2.2)$$

Hodaň, Matas und Obdržálek gelangen jedoch zu dem Schluss, dass die Eignung der Objektberflächenorientierung der wichtigste Indikator für die Qualität der geschätzten Objektpose ist. Daher ist die Verwendung von Fehlerfunktionen zu empfehlen, welche das 3D-Modell des Objektes berücksichtigen [32].

Eine solche Metrik, welche in abgewandelter Form immer noch verwendet wird, stellen Xiang u. a. im Rahmen der Entwicklung der Methode *PoseCNN* Methode. Die Autoren verwenden

die *Average Distance* (ADD, deutsch: Durchschnittliche Entfernung) zur Bewertung. Diese verwendet die Ground-Truth-Rotation R und Translation T sowie die geschätzte Rotation \tilde{R} und Translation \tilde{T} (das Ergebnis der 6D-Posenbestimmung). Die Average Distance berechnet den Mittelwert der paarweisen Abstände der Punkte des 3D-Modells zwischen der Ground-Truth-Pose und der geschätzten 6D-Pose ([35]nach [33]):

$$\text{ADD} = \frac{1}{m} \sum_{x \in M} \|(Rx + t) - (\tilde{R}x + \tilde{T})\| \quad (2.3)$$

Bei symmetrischen Objekten ist die Übereinstimmung der Punkte auf den 3D-Modellen in einigen Ansichten nicht eindeutig. Daher wird der durchschnittliche Abstand anhand des nächstgelegenen Punktes x_2 berechnet:

$$\text{ADD-S} = \frac{1}{m} \sum_{x_1 \in M} \min_{x_2 \in M} \|(Rx_1 + t) - (\tilde{R}x_2 + \tilde{T})\| \quad (2.4)$$

Somit kann die Orientierung eines Objektes entlang der Symmetrieachsen beliebig sein, ohne dass dies eine negative Auswirkung auf die Bewertung hat. Eine 6D-Pose gilt als korrekt, wenn der jeweilige Durchschnitt kleiner ist als ein zuvor festgelegter Schwellwert. Je nach Datensatz liegt dieser Wert bei 10 % des 3D-Modelldurchmessers [35].

Der Status quo der Bewertung von 6D-Posenbestimmung wird durch die in der *BOP-Challenge* (engl.: Benchmark for 6D Object Pose Estimation) verwendeten Metriken dargestellt [36, 37, 38, 39, 14]. Die BOP-Challenge verwendet mehrere *Pose-Fehlerfunktionen* zur Berechnung der Genauigkeit einer Methode zur 6D-Posenbestimmung. Der Fehler einer geschätzten Pose $\hat{\mathbf{P}}$ im Vergleich zur tatsächlichen Pose $\bar{\mathbf{P}}$ eines Objektmodells M wird durch drei Pose-Fehlerfunktionen bestimmt:

Visible Surface Discrepancy (VSD, dt. Abweichung der sichtbaren Oberflächen) ([38] nach [32]):

$$e_{\text{VSD}}(\hat{D}, \bar{D}, \hat{V}, \bar{V}, \tau) = \text{avg}_{p \in \hat{V} \cup \bar{V}} \begin{cases} 0 & \text{wenn } p \in \hat{V} \cap \bar{V} \wedge |\hat{D}(p) - \bar{D}(p)| < \tau \\ 1 & \text{sonst} \end{cases} \quad (2.5)$$

\hat{D} und \bar{D} stehen für Distanzkarten, die durch das Rendern des Objektmodells M in der geschätzten Pose $\hat{\mathbf{P}}$ bzw. der Referenzpose $\bar{\mathbf{P}}$ erhalten werden. Diese Distanzkarten werden mit der Distanzkarte D_I des Testbildes I verglichen, um die Sichtbarkeitsmasken \hat{V} und \bar{V} zu er-

halten, also die Pixelbereiche, in denen das Modell M im Bild I sichtbar ist. Der Parameter τ gibt die Toleranz für die Fehlanpassung an. Die Funktion `avg` berechnet den Durchschnitt für unterschiedliche Werte von τ . Im Unterschied zu früheren Ansätzen wurde die Schätzung der Sichtbarkeitsmasken geändert, indem ein Objekt als sichtbar an Pixeln angenommen wird, an denen keine Tiefenmessungen vorhanden sind. Diese Änderung ermöglicht die Bewertung der Posen von glänzenden Objekten, deren Oberflächen nicht immer im Tiefenbildkanal erfasst werden. Die Fehlerfunktion VSD behandelt Posen, die in ihrer Form ununterscheidbar sind als gleichwertig (Farbe wird nicht berücksichtigt), indem es nur die Fehlanpassung des sichtbaren Teils der Objektoberfläche misst [38].

Maximum Symmetry-Aware Surface Distance (MSSD, deutsch: Maximaler symmetriesensitiver Oberflächenabstand):

$$e_{\text{MSSD}}(\hat{\mathbf{P}}, \bar{\mathbf{P}}, S_M, V_M) = \min_{s \in S_M} \max_{x \in V_M} \|\hat{\mathbf{P}}\mathbf{x} - \bar{\mathbf{P}}\mathbf{S}\mathbf{x}\|_2 \quad (2.6)$$

In der Menge S_M befinden sich die globalen Symmetrietransformationen des Objektmodells M und V_M steht für die Menge der Punkte (engl.: vertices) des Modells. Für die robotergestützte Handhabung ist die maximale Entfernung zwischen den Punkten des Modells von Bedeutung, da die Abweichung der Oberflächen mit der Erfolgswahrscheinlichkeit eines Greifvorgangs zusammenhängt. MSSD basiert auf ADD (Gleichung (2.3)), jedoch ist die maximale Entfernung ist weniger abhängig von der Geometrie des Objektmodells und der Abtastdichte seiner Oberfläche [38].

Maximum Symmetry-Aware Projection Distance (MSPD, deutsch: Maximaler symmetriesensitiver Projektionsabstand):

$$e_{\text{MSPD}}(\hat{\mathbf{P}}, \bar{\mathbf{P}}, S_M, V_M) = \min_{s \in S_M} \max_{x \in V_M} \|\text{proj}(\hat{\mathbf{P}}\mathbf{x}) - \text{proj}(\bar{\mathbf{P}}\mathbf{S}\mathbf{x})\|_2 \quad (2.7)$$

Die Funktion `proj` stellt die 2D-Projektion dar, wobei das Ergebnis in Pixeln angegeben wird. Die Bedeutung der anderen Symbole entspricht denen in MSSD. MSPD berücksichtigt globale Objektsymmetrien und ersetzt den Durchschnitt durch die maximale Distanz, um die Robustheit gegenüber der Geometrie und Abtastung des Objektmodells zu erhöhen. Da MSPD keine Ausrichtung entlang der optischen z -Achse bewertet und nur die wahrnehmbare Abweichung misst, ist es besonders relevant für Anwendungen in der Augmented Reality [38].

Die drei Fehlerfunktionen werden anschließend in dem **Average Recall** (deutsch: Durchschnittli-

cher Rückruf, sinngemäß Wiedererkennung) zusammengefasst. Eine geschätzte Pose gilt als korrekt in Bezug auf eine Pose-Fehlerfunktion e , wenn $e < \theta_e$ ist, wobei $e \in e_{VSD}, e_{MSSD}, e_{MSPD}$ und θ_e ein Schwellenwert für die Korrektheit ist. Der Anteil der annotierten Objektinstanzen, für den eine korrekte Pose geschätzt wird, wird als Recall (sinngemäß Wiedererkennung) bezeichnet. Der Average Recall in Bezug auf eine die Pose-Fehlerfunktion e , bezeichnet als AR_e , wird definiert als der Durchschnitt der Recall-Raten, die für mehrere Einstellungen des Schwellenwerts θ_e berechnet werden, sowie für mehrere Einstellungen der Fehlertoleranz τ im Fall von e_{VSD} . Insbesondere ist AR_{VSD} der Durchschnitt der Recall-Raten, die für τ von 5% bis 50% des Objektdurchmessers in Schritten von 5% und für θ_{VSD} von 0.05 bis 0.5 in Schritten von 0.05 berechnet wird. AR_{MSSD} ist der Durchschnitt der Recall-Raten, die für θ_{MSSD} von 5% bis 50% des Objektdurchmessers in Schritten von 5% berechnet wird. Schließlich ist AR_{MSPD} der Durchschnitt der Recall-Raten, die für θ_{MSPD} von $5r$ bis $50r$ in Schritten von $5r$ berechnet wird, wobei $r = w/640$ und w die Bildbreite in Pixeln ist.

Die Evaluierung der Präzision einer Methode hinsichtlich eines Datensatzes D erfolgt schließlich gemäß der folgenden Formel:

$$AR_D = \frac{(AR_{VSD} + AR_{MSSD} + AR_{MSPD})}{3} \quad (2.8)$$

In der Folge lässt sich die Gesamtgenauigkeit einer Methode auf allen Datensätzen der BOP-Classic-Core Challenge (vgl. [40]) mit AR_{Core} aus dem Durchschnitt aller Werte AR_D berechnen.

Diese sehr indirekte und komplexe mathematische Berechnung des Recalls verdeutlicht ein grundlegendes Problem, das auftritt, wenn die Eignung einer Methode für den praktischen Einsatz in der Industrie beurteilt werden soll: Angenommen, die Pose eines sphärischen Objekts mit einem Durchmesser von 100 mm wird stets mit einer Abweichung von 5 mm geschätzt. Dies würde dennoch zu einem Wert von $e_{VSD} = 0$ und somit zu einem $AR_{VSD} = 1$ führen. Die entsprechende Methode hätte somit eine perfekte Bewertung der Abweichung der sichtbaren Oberflächen, obwohl diese stets um 5 mm abweichen. Diese Abweichung ist jedoch unzureichend, um die Prozesssicherheit industrieller Anwendungen zu gewährleisten.

Somit lässt sich feststellen, dass unterschiedliche Anforderungen an die Bewertung einer Methode zur 6D-Posenbestimmung bestehen, wenn diese hinsichtlich ihrer Eignung für den industriellen Einsatz beurteilt werden soll.

2.4.2. Datensätze für die 6D-Posenbestimmung

Wie im vorherigen Teil dieser Arbeit mehrfach erwähnt, erfolgt die Bewertung von Methoden zur 6D-Posenbestimmung anhand verschiedener Datensätze. Diese Datensätze wurden in der Vergangenheit auch zum Training der Methoden genutzt. Ein Kernmerkmal dieser Datensätze ist, dass sie Trainings- und Testbilder der Objekte enthalten, die mit den tatsächlichen 6D-Posen der Objekte, der sogenannten Ground Truth, versehen sind. Die Daten werden in einem einheitlichen Format bereitgestellt, wobei jeder Datensatz 3D-Modelle sowie RGB-Bilder und gegebenenfalls auch RGB-D-Bilder umfasst. Die 3D-Objektmodelle können manuell oder mithilfe von Systemen zur 3D-Oberflächenrekonstruktion erstellt worden sein, weshalb die Qualität der 3D-Modelle stark variieren kann. Umfang, Art der Objekte, Aufbau der Szenen und weitere Eigenschaften können dabei ebenfalls stark unterschiedlich ausfallen.

Einen besonderen Stellenwert bei der Bewertung nimmt die BOP-Challenge ein, wie bereits in Abschnitt 2.4.1 erwähnt. Die **BOP-Core Challenge** umfasst sieben Datensätze, siehe Abbildung 16. Die Datensätze T-LESS, TUD-L und YCB-V umfassen ausschließlich Trainingsbilder, die unter Verwendung echter Szenen erstellt wurden. Die restlichen Datensätze beinhalten zudem Trainingsbilder, die durch OpenGL-Rendering der 3D-Modelle auf schwarzem Hintergrund erzeugt wurden. Die Testbilder wurden in Szenen mit variierender Komplexität aufgenommen, wobei häufig Unordnung und partielle Verdeckung zu beobachten sind [38]. Die Evaluierung einer Methode zur 6D-Posenbestimmung auf den Datensätzen dieser Challenge definiert den aktuellen Standard in der 6D-Posenbestimmung.



Abbildung 16: Die Datensätze der BOP Core Challenge: T-LESS, ITODD und YCB-V sind beschriftet, adaptiert aus [38]

Im Jahr 2024 haben die Verantwortlichen der BOP-Challenge drei neue Datensätze eingeführt (siehe Abbildung 17), die eine eigenständige Herausforderung bieten. Die **BOP-H3 Challenge** legt einen starken Fokus auf Hand-Objekt-Interaktionen und soll zukünftig die Bewertung entsprechender Methoden in realistischen Szenarien erleichtern [40]. Die Challenge umfasst die folgenden drei Datensätze:

- **HOT3D** [41] bietet Multi-View-RGB- und monochrome Bildströme von Interaktionen mit 33 starren Objekten. Erfasst wurden die Daten mit den Meta-Geräten Project Aria und Quest 3. Die Bilder sind mit präzisen 3D-Posen und -Formen annotiert, die mithilfe eines Motion-Capture-Systems und optischer Marker gewonnen wurden. HOT3D wird auch für die HANDS Challenge 2024 verwendet.
- **HOPEv2** [42] ist ein Datensatz für die robotische Manipulation, bestehend aus 28 Spielzeug-Gegenständen. Ursprünglich in 50 unaufgeräumten Szenarien mit verschiedenen Beleuchtungen erfasst, wurde der Datensatz für die BOP Challenge 2024 um zusätzliche Testbilder aus sieben weiteren Szenen erweitert. (Der HOPE Datensatz wird im Folgenden noch kurz näher beschrieben.)
- **HANDAL** [43] enthält greifbare Objekte wie Hämmer und Tassen, aufgenommen aus mehreren Blickwinkeln in unaufgeräumten Szenarien. Für die BOP Challenge 2024 wurden zu den ursprünglichen 212 Objekten aus 17 Kategorien zusätzliche Testbilder hinzugefügt; der Fokus liegt jedoch auf 40 Objekten aus sieben Kategorien, die qualitativ hochwertige CAD-Modelle besitzen.



Abbildung 17: Datensätze der neuen BOP H3 Challenge mit Fokus auf Hand-Interaktion, adaptiert von [40]

Im Folgenden sollen die Datensätze YCB-V, T-LESS, ITODD und HOPE näher betrachtet werden, um exemplarisch die Motivationen für die Objektauswahl vorzustellen und den Prozess zur Erstellung der jeweiligen Datensätze kurz zu erläutern.

Als Beispiel für einen klassischen Datensatz mit texturierten Objekten dient der **YCB-V** (YCB-Video) Datensatz, welcher von Xiang u. a. im Rahmen ihrer Arbeit zur Methode PoseCNN erstellt wurde. Dieser Datensatz basiert auf dem YCB (Yale-CMU-Berkeley) Datensatz von Calli u. a. [44]. YCB-V umfasst jedoch nur 21 der ursprünglich 75 Objekte, siehe Abbildung 18. Ausgeschlossen wurden dabei nicht starre Objekte sowie Objekte, die aufgrund anderer Eigenschaften zum damaligen Stand nicht für die 6D-Posenbestimmung geeignet waren. Die Videos für den YCB-V Datensatz wurden mit einer Asus Xtion Pro Live RGB-D-Kamera aufgenommen. Der vollständige Datensatz umfasst 133.827 Bilder, welche für das Training der PoseCNN-Methode verwendet wurden [35].



Abbildung 18: Die 21 Objekte des YCB-V Datensatzes [35]

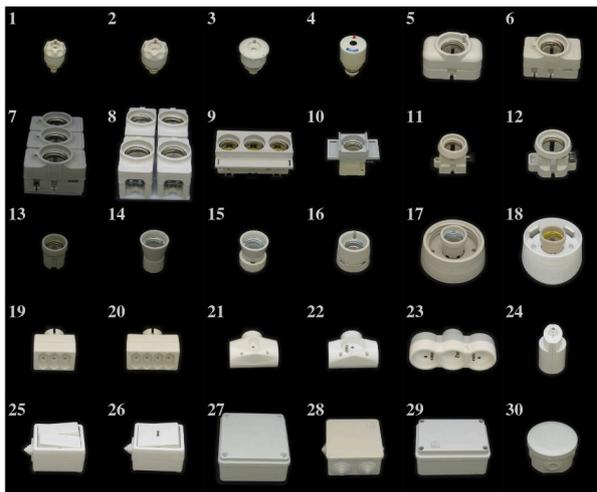
Um die manuelle Annotation aller Videoframes zu vermeiden, spezifizieren Xiang u. a. die Positionen der Objekte manuell nur im ersten Frame jedes Videos. Anhand von *Signed Distance Function*-Darstellungen² jedes Objekts wird die Position im ersten Tiefen-Frame verfeinert. Anschließend wird die Kameratrajektorie initialisiert, indem die Objekthaltungen relativ zueinander fixiert und die Objektkonfiguration durch das Tiefenvideo verfolgt werden. Schließlich werden die Kameratrajektorie und die relativen Objekthaltungen in einem globalen Optimierungsschritt verfeinert [35].

Der YCB-V Datensatz hat sich als besonders erfolgreich erwiesen, da er überwiegend aus Objekten besteht, die relativ leicht verfügbar sind. Diese Objekte sind häufig in den meisten Supermärkten in den USA zu finden³. Dies ermöglicht es Forschenden, Methoden, die auf diesem Datensatz trainiert wurden, auch in eigenen experimentellen Versuchen anzuwenden.

²Eine Signed Distance Function ist eine mathematische Funktion, die jedem Punkt im Raum einen Wert zuweist, der den Abstand dieses Punktes zu einer definierten Oberfläche angibt, wobei das Vorzeichen des Wertes angibt, ob der Punkt innerhalb (negativ) oder außerhalb (positiv) der Oberfläche liegt.

³Die YCB-V Objekte können auch außerhalb der USA bestellt werden: <https://www.ycbbenchmarks.com/>

Hodaň u. a. stellen in [45] den Datensatz **T-LESS** (Texture-less, deutsch: Texturlos) vor. Der Datensatz umfasst dreißig für die Industrie relevante Objekte ohne signifikante Textur und ohne kennzeichnende Farb- oder Reflexionseigenschaften. Die Objekte stammen aus dem Bereich der Elektrikinstallation, wobei jedoch eine gewisse Einschränkung der freien Verfügbarkeit zu verzeichnen ist. Die Objekte weisen Symmetrien und gegenseitige Ähnlichkeiten in Form und/oder Größe auf. Im Vergleich zu anderen Datensätzen besitzt dieser die einzigartige Eigenschaft, dass einige Objekte Teile anderer sind. Für jedes Objekt werden zwei Arten von 3D-Modellen bereitgestellt, nämlich ein manuell erstelltes CAD-Modell und ein halbautomatisch rekonstruiertes Modell. Die Testbilder stammen aus zwanzig Testszenen mit unterschiedlicher Komplexität, die von einfachen Szenen mit einigen isolierten Objekten bis hin zu sehr anspruchsvollen Szenen mit mehreren Instanzen verschiedener Objekte und einer hohen Menge an Unordnung und Verdeckung reichen [45].



(a) Die 30 Objekte



(b) 20 Testbilder mit überlagerter Ground-Truth

Abbildung 19: Objekte und Szenen aus dem T-LESS Datensatz [45]

Die Bilder wurden aus einer systematisch abgetasteten Sichtsphäre um das Objekt/die Szene herum aufgenommen und sind mit genauen 6D-Posen aller modellierten Objekte versehen. Hierfür verwenden Hodaň u. a. den in Abbildung 20 dargestellten Aufbau.

Die Bilder für den T-LESS Datensatz wurden durch ein systematisches Aufnahmeverfahren erstellt. Dieses umfasst Aufnahmen von 85° bis -85° in 10° Schritten sowie den gesamten Azimutbereich in 5° Schritten. Ansichten der oberen und unteren Hemisphären wurden separat akquiriert, indem das Objekt zwischendurch umgedreht wurde. Insgesamt wurden pro Objekt und Sensor 1296 Trainingsbilder aufgenommen. Testszenen wurden von einer Ansichtshalbkugel in 10° Schritten von 75° bis 15° aufgenommen, was zu insgesamt 504 Testbildern pro Szene und Sensor führt.

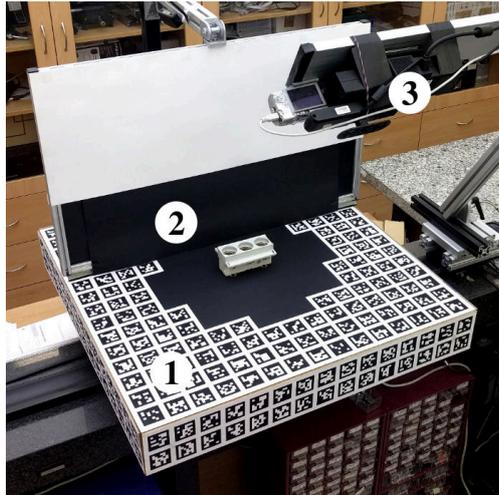


Abbildung 20: Aufbau zur Aufnahme der T-LESS Daten: 1) Drehtisch mit Markern, 2) Hintergrund für Trainingsdaten, 3) Sensoren an neigbarer Vorrichtung [45]

Die Ground-Truth-Annotation wurde anhand der über die Marker auf dem Drehtisch geschätzten Sensorpositionen durchgeführt. Zur Erhöhung der Genauigkeit wurden die Objektmodelle in mehrere hochauflösende Szenenbilder einer Canon-Kamera gerendert, um Fehlanpassungen zu identifizieren. Die Positionen wurden manuell verfeinert, bis eine zufriedenstellende Übereinstimmung der Renderings mit den Szenenbildern erreicht war. Anschließend wurden die endgültigen Positionen anhand der bekannten Transformationen von Kamera- zu Drehtischkoordinaten auf alle Testbilder übertragen [45].

Eine weitere Besonderheit unter den Datensätzen zur Evaluierung der 6D-Posenbestimmung stellt der **ITODD** Datensatz dar (Industrial 3D Object Detection Dataset, deutsch: Industrieller 3D-Objekt-Erkennungsdatensatz). Dieser Datensatz legt einen starken Fokus auf Objekte, Szenarien und Anforderungen, die für industrielle Umgebungen realistisch sind. Dies steht im Gegensatz zu anderen Datensätzen, die häufig Szenarien aus dem Alltag oder mobilen Robotikumgebungen darstellen. Aufgrund dieser Ausrichtung modelliert der Aufbau zur Datensatzaufnahme auch Aufgaben wie das industrielle Bin-Picking und die Objektinspektion [46]. Eine Abbildung aller Objekte des Datensatzes findet sich in Anhang A.1.

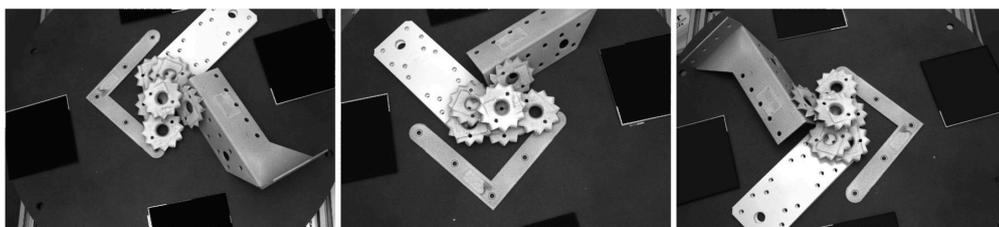


Abbildung 21: Exemplarische Aufnahme des ITODD Datensatzes, drei unterschiedliche Blickwinkel der 2D Kamera in Graustufen [46]

Für die Aufnahme des Datensatzes wurden zwei industrielle 3D-Sensoren und drei hochauflösende Graustufenkameras verwendet (siehe Abbildung 22). Diese erfassen die Szene aus verschiedenen Blickwinkeln, was die Bewertung von Methoden ermöglicht, die auf Mehrfachansicht basieren (vgl. Abbildung 21).

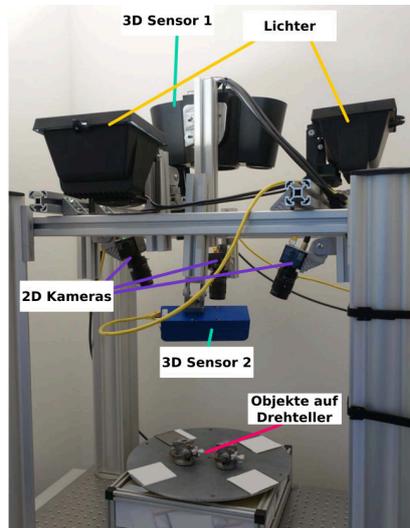


Abbildung 22: Aufbau zur Datenaufnahme des ITODD Datensatzes, adaptiert aus [46]

Die Ground-Truth wurde mittels eines halbmanuellen Ansatzes auf Basis der 3D-Daten eines industriellen 3D-Sensors annotiert. Jede Objektinstanz wurde zunächst grob von Hand segmentiert, gefolgt von mehreren Iterationen des manuellen Ausführens des Iterative Closest Point (ICP) Algorithmus, bis sowohl eine gute Punktzahl als auch ein visuell korrektes Ergebnis erzielt wurde. Die entsprechenden Ground-Truth Posen wurden auf die durch die Drehung des Drehtisches erhaltenen Szenen unter Verwendung der kalibrierten Drehtischpositionen übertragen [46].

Der **HOPE** (Household Objects for Pose Estimation) Datensatz besteht aus RGB-D-Bildern und Videosequenzen mit annotierter Ground-Truth für 28 Spielzeug-Lebensmittelobjekte. Diese Spielzeug-Lebensmittelobjekte sind leicht erhältlich und haben die ideale Größe sowie das ideale Gewicht für die robotergestützte Manipulation, vgl. Abbildung 23.



(a) Foto sämtlicher Objekte



(b) Synthetisches Rendering der 3D Modelle

Abbildung 23: Die Objekte des HOPE Datensatzes [42]

Die Objekte werden in 50 Szenen aus 10 unterschiedlichen Haushalts- und Büro-Umgebungen präsentiert. Für jede Szene wurden bis zu 5 Beleuchtungsvariationen erfasst, einschließlich Gegenlicht und seitlichem direkten Licht mit Schattenwürfen, vgl. Abbildung 24.



Abbildung 24: Unterschiedliche Umgebungen im HOPE Datensatz, adaptiert aus [42]

Des Weiteren haben die Szenen unterschiedliche Objektanordnungen, die zu verschiedenen Graden der Objektverdeckung führen, vgl. Abbildung 25.



Abbildung 25: Unterschiedliche Objektanordnungen im HOPE Datensatz, adaptiert aus [42]

Zur Annotation der Daten mit der Ground-Truth haben die Autoren zwei unterschiedliche Verfahren genutzt. Im Perspective-n-Point (PnP)-Verfahren⁴ wählt der Annotator entsprechende Punkte sowohl auf der 2D-Texturkarte des Objektmodells als auch im Bild, in dem das Objekt erscheint, aus. Alternativ werden in der RGB-D-Version Beziehungen zwischen dem 3D-texturierten Modell und der RGB-D-Tiefenkarte hergestellt. Das jeweils bessere Ergebnis wurde manuell ausgewählt und, falls notwendig, manuell weiter verfeinert [42].

Zusammenfassend lässt sich feststellen, dass die Annotation der aufgenommenen Daten mit der Ground-Truth für die 6D-Posenbestimmung einen erheblichen Aufwand erfordert und oft eine erhebliche Menge manueller Eingriffe notwendig macht. Diese aufwändige Prozesskette beeinflusst nicht nur die Genauigkeit der Ergebnisse, sondern erhöht vor allem den erforderlichen Arbeitsaufwand erheblich. Im weiteren Verlauf dieser Arbeit wird ein Verfahren entwickelt und vorgestellt, welches diesen bedeutenden Nachteil adressiert und eine effizientere sowie weniger fehleranfällige Datenannotation ermöglicht.

⁴PnP ist eine grundlegende Aufgabe in der Computervision. Dabei wird die Position und Orientierung einer Kamera im Raum bestimmt, wenn die 3D-Koordinaten von mindestens n Punkten in der Szene und die 2D-Projektionspunkte dieser 3D-Punkte im Kamerabild bekannt sind.

2.4.3. Sensoren für die 6D-Posenbestimmung

Im Folgenden werden die im weiteren Verlauf der Arbeit verwendeten Sensoren zur Aufnahme annotierter Testdaten vorgestellt. Hierbei handelt es sich um die Intel RealSense D435i sowie die Luxonis OAK-D Pro.

Beide Sensoren lassen sich mit einem Preis von circa 370 € prinzipiell dem Low-Cost-Segment zuordnen, das heißt, ihr Preis liegt um ein Vielfaches unter dem, was ähnliche Sensoren aus dem industriellen Bereich kosten. Des Weiteren sind beide Sensoren mit einer Inertial Measurement Unit (IMU) ausgestattet. Die IMU ermöglicht durch die Messung von Beschleunigungen und Drehraten präzise Informationen über die Orientierung und Bewegung des Geräts. Durch die Fusionierung dieser Daten mit visuellen Sensorinformationen kann eine deutliche Verbesserung der Tracking-Genauigkeit und -Stabilität erzielt werden. Ein tabellarischer Vergleich der beiden Sensoren findet sich in Tabelle 2.

	Intel RealSense D435i	Luxonis OAK-D Pro
<i>RGB-Sensor</i>		
Auflösung	1920x1080 Pixel	4056x3040 Pixel
Bildrate	30 FPS	60 FPS
Fokusbereich (ab)	keine Angabe	8 cm
FoV (H x V)	69° × 42°	66° × 54°
Verschlussstyp	Rolling	Rolling
<i>Stereo Tiefen-Sensoren</i>		
Auflösung	1280x720 Pixel	1280x800 Pixel
Fokusbereich (ab)	ca. 28 cm	19,6 cm
FoV (H x V)	keine Angabe	82° × 57°
Verschlussstyp	Rolling	Rolling
Abmessungen (H x B x T)	25mm × 90mm × 25mm	29,5mm × 97mm × 23,1mm

Tabelle 2: Vergleich von Intel RealSense D435i und Luxonis OAK-D Pro (Autofokus) [47, 48]

Beide Sensoren weisen ein vergleichbares Field of View (FoV) sowie einen relativ ähnlichen Fokusbereich auf. Die Intel RealSense D435i verfügt über eine geringere Auflösung als die Luxonis OAK-D Pro. Sie bietet jedoch den Vorteil einer breiten Marktakzeptanz, eines umfangreichen und robusten Software-Entwicklungskits sowie einer aktiven Entwickler-Community. In vielen Bereichen stellen die Sensoren der RealSense-Serie den Standard dar. Die OAK-D Pro bietet neben der höheren Auflösung den Vorteil, dass das Gerät über eine integrierte Recheneinheit, den Robotics Vision Core 2, verfügt. Der integrierte Recheneinheit ermöglicht die direkte Verwendung von KI-gestützten Modellen auf dem Gerät. Standardmäßig stehen verschiedene Modelle für die Objekterkennung zur Verfügung, die mit einer Auflösung von bis zu 640x640 Pixeln und einer Bildrate von 30 FPS betrieben werden können [49].

2.5. Aktuelle Methoden für die 6D-Posenbestimmung von neuartigen Objekten

In diesem Kapitel erfolgt zunächst eine Betrachtung der generellen Entwicklung von Methoden zur 6D-Posenbestimmung in den vergangenen Jahren. Zu diesem Zweck werden die Bestenlisten der bereits mehrfach erwähnten BOP Challenge [36] unter verschiedenen Aspekten analysiert. Im Anschluss erfolgt eine Vorstellung der in der weiteren Arbeit verwendeten Methoden MegaPose, SAM6D und FoundationPose.

Die Entwicklung des Average Recall (vgl. Abschnitt 2.4.1), den die jeweils besten Methoden seit der Vorstellung der BOP-Challenge auf den Datensätzen der BOP-Core Challenge (vgl. Abschnitt 2.4.2) erreichen, ist in Abbildung 26 dargestellt.

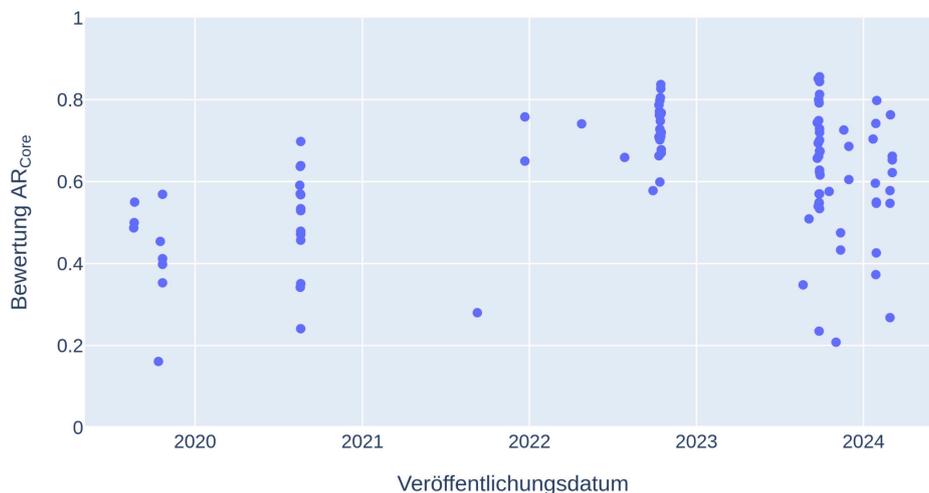


Abbildung 26: Entwicklung der Methoden zur 6D-Posenbestimmung, Daten aus [36]

Zunächst lässt sich an dieser Abbildung sehr gut erkennen, dass eine Vielzahl der Methoden jeweils zu den jährlichen BOP-Challenges eingereicht wird⁵. Weiterhin ist ersichtlich, dass sich die Methoden im Laufe der Jahre signifikant verbessert haben und dass allgemein mehr Methoden bzw. Ergebnisse bei den Challenges eingereicht werden.

Es scheint, dass im Jahr 2023/2024 mehrere Ergebnisse eingereicht wurden, die eine geringere Qualität aufweisen als vorherige Methoden. Diese Beobachtung ist auf die Einführung einer neuen Challenge zur Posenbestimmung von neuartigen Objekten im genannten Zeitraum zurückzuführen [14], vgl. Abbildung 27.

⁵Die BOP Challenges fallen jeweils mit den größten und renommiertesten Konferenzen für Computer Vision zusammen (ICCV, ECCV, CVPR, etc.).

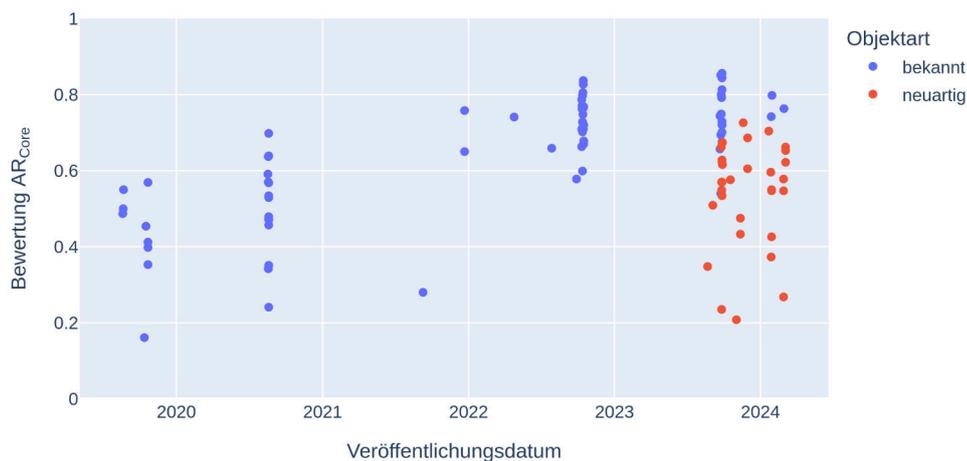


Abbildung 27: Methoden zur 6D Posenbestimmung nach bekannten und neuartigen Objekten, Daten aus [36]

Wie zu erkennen ist, handelt es sich bei den entsprechenden Methoden um solche, die die Posenbestimmung von neuartigen Objekten ermöglichen (vgl. Abschnitt 2.4). Diese Methoden bieten den Vorteil, dass kein objektspezifisches Training für die Posenbestimmung notwendig ist, was den Aufwand für die Implementierung solcher Methoden erheblich reduziert [14]. Allerdings besteht ein Trade-off bei der erreichbaren Genauigkeit, wengleich aktuelle Methoden diesen teilweise reduzieren können.

Die Unterscheidung der erreichbaren Genauigkeit in Abhängigkeit von der verwendeten Bildart, also ob es sich um reine Tiefeninformationen, Farbbilder oder kombinierte Farb- und Tiefenbilder handelt, ist in Abbildung 28 dargestellt.

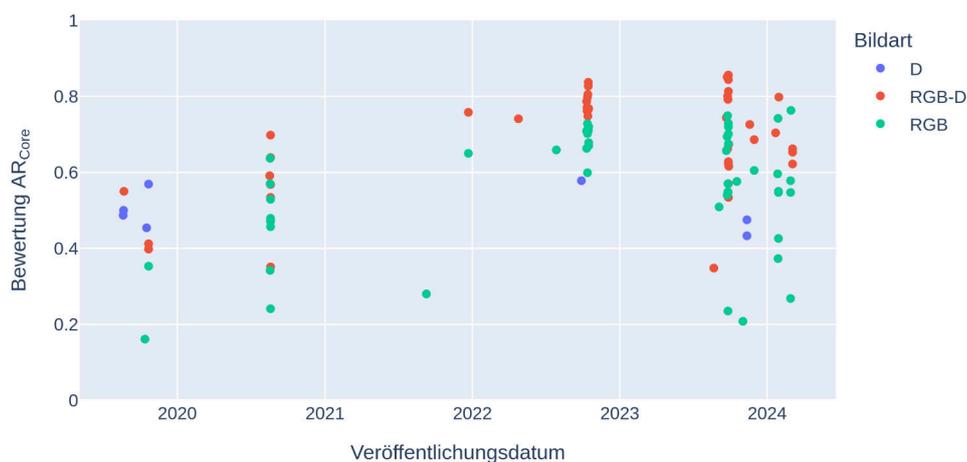


Abbildung 28: Methoden zur 6D-Posenbestimmung nach RGB, RGB-D und reinen Tiefendaten, Daten aus [36]

Zu Beginn der BOP Challenges stellte die Generalisierung eine große Herausforderung dar, da im Vergleich zu heute noch relativ kleine Datensätze verwendet wurden. Dies konnte als einer der Gründe angesehen werden, warum rein tiefenbasierte Methoden (D) damals noch sehr gut abschnitten, da sie weniger von diesem Problem betroffen waren. Mit der Verfügbarkeit umfangreicherer Trainingsdaten sind diese Ansätze heute nur noch für spezifische Anwendungsfälle relevant. Wie bereits kurz in Abschnitt 2.4 beschrieben, bieten RGB-D-Methoden durch die zusätzlichen Tiefeninformationen im Vergleich zu reinen RGB-Methoden den Vorteil, prinzipiell höhere Genauigkeiten zu erreichen.

Ein weiterer sehr interessanter Vergleich ist die erreichbare Genauigkeit der Posenbestimmung in Abhängigkeit von spezifischen Objekteigenschaften. Aus industrieller Sicht ist hierbei insbesondere die Posenbestimmung texturloser Bauteile relevant. Einen Vergleich hierzu zeigt Abbildung 29, der die jeweils erreichten Average Recall Werte auf dem YCB-V und T-LESS Datensatz darstellt (vgl. Abschnitt 2.4.2).



Abbildung 29: Methoden zur 6D-Posenbestimmung nach Objektdatensätzen YCB-V und T-LESS (ab August 2023), Daten aus [36]

Wie zu erkennen ist, werden auf texturierten Objekten, wie sie größtenteils im YCB-V-Datensatz zu finden sind, prinzipiell höhere Genauigkeiten erreicht. RGB-D-Methoden helfen zwar dabei, diese Lücke zumindest teilweise zu schließen. Es soll jedoch verdeutlicht werden, dass die Erkennung von texturlosen und insbesondere von metallischen Bauteilen immer noch eine Herausforderung darstellt.

Im Kontext der Posenbestimmung von texturlosen Objekten sind Methoden, die Surface Encoding verwenden, besonders vielversprechend, wie beispielsweise in [50] beschrieben. Da diese Methoden jedoch noch nicht für neuartige Objekte verwendet werden können, finden sie im weiteren Verlauf

der Arbeit keine weitere Erwähnung.

Ein weiterer sehr interessanter und besonders für industrielle Anwendungsfälle gewinnbringender Ansatz ist die Möglichkeit der Multiview-Posenbestimmung. Die Verwendung mehrerer Ansichten ermöglicht es einerseits, Herausforderungen wie partielle (Selbst-)Verdeckungen zu mildern, und andererseits, durch mehrere Posenbestimmungen die Robustheit gegen etwaige Fehler zu erhöhen, vgl. [51]. Aktuelle Methoden basieren jedoch nahezu ausschließlich auf Single-View-Ansätzen, weshalb auch auf diese nicht weiter eingegangen wird.

Ein Vergleich der drei Methoden, die im weiteren Verlauf der Arbeit genutzt werden, aus den BOP-Challenges findet sich in Abbildung 30.

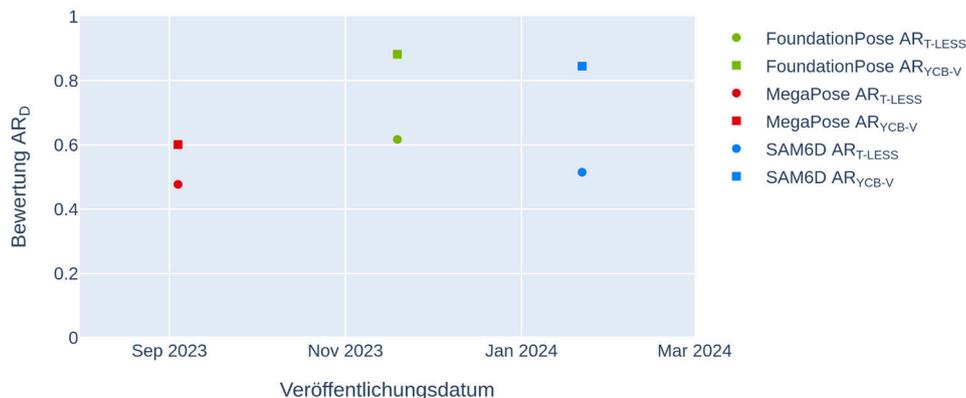


Abbildung 30: Verwendete Methoden zur 6D-Posenbestimmung im Vergleich für die Datensätze YCB-V und T-LESS, Daten aus [36]

Wie ersichtlich ist, erzielt FoundationPose die höchsten Genauigkeiten unter den drei Methoden. Die Ergebnisse von MegaPose sind jedoch insbesondere in Anbetracht der Tatsache, dass diese ohne die Verwendung von Tiefendaten erreicht wurden, ebenfalls sehr gut. Bei SAM6D ist ein vergleichsweise großer Unterschied zwischen der erreichten Genauigkeit auf den texturierten Objekten des YCB-V-Datensatzes und den texturlosen Objekten des T-LESS-Datensatzes festzustellen. Insgesamt sind jedoch auch diese Ergebnisse bemerkenswert, vor allem, da SAM6D auch die Region of Interest selbst bestimmt. Ein ausführlicher tabellarischer Vergleich der Methoden basierend auf der BOP-Challenge findet sich in Anhang A.2. Im Folgenden sollen einige Besonderheiten der verwendeten Methoden kurz vorgestellt werden.

2.5.1. MegaPose

Labbé u. a. stellen mit MegaPose eine Methode für die 6D-Posenbestimmung vor, die in der Lage ist, ausschließlich auf Basis von RGB-Bildern eine Posenbestimmung durchzuführen⁶. Hierzu benötigt die Methode lediglich ein CAD-Modell des Objektes und eine Region of Interest. Die Autoren verwenden dafür einen zweistufigen Ansatz (vgl. Abbildung 31).

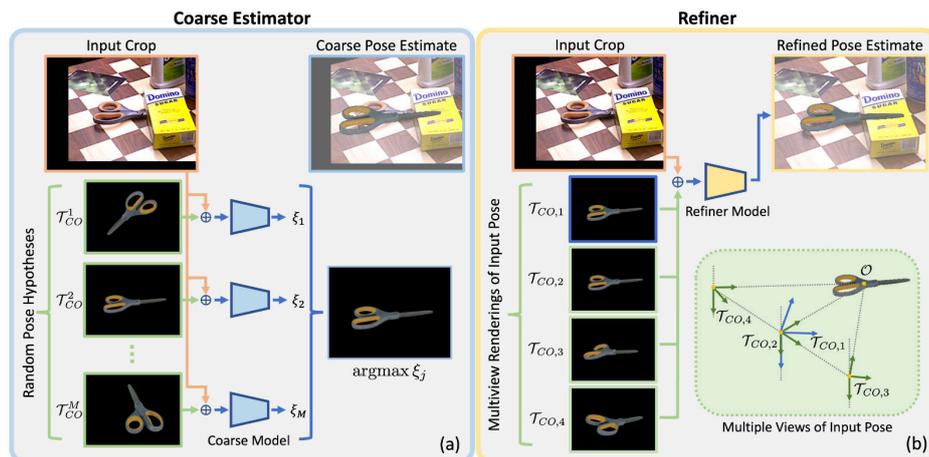


Abbildung 31: Ablauf der groben und feinen Posenbestimmung mit MegaPose [52]

Der Coarse Estimator (deutsch: grobe Pose-Schätzer) hat das Ziel, eine anfängliche Schätzung $\tau_{CO,coarse}$ der Pose zu liefern. Diese Pose muss ausreichend präzise sein, damit sie durch nachfolgende Verfeinerungsverfahren weiter optimiert werden kann. Um eine Generalisierung auf neue Objekte zu erreichen, wird ein innovativer, auf Klassifikation basierender Ansatz verwendet. Dieser Ansatz vergleicht beobachtete und gerenderte Bilder des Objekts in verschiedenen Posen und wählt das gerenderte Bild aus, dessen Objektpose die größte Übereinstimmung mit der beobachteten Pose aufweist.

Der Refiner (deutsch: Verfeinerer) generiert eine aktualisierte Schätzung der 6D-Pose. Beginnend mit der groben initialen Pose-Schätzung $\tau_{CO,coarse}$ lässt sich der Verfeinerer iterativ anwenden, um schrittweise eine verbesserte Pose-Schätzung $\tau_{CO}^{k=1}$ zu erreichen, wobei k die Anzahl der Iterationen ist [52].

Für das Training benötigen sowohl der Coarse Estimator als auch der Refiner Bilder mit Ground-Truth sowie 3D-Modelle dieser Objekte. Um eine Methode auf neue Objekte zu verallgemeinern, wird ein besonders großer Datensatz mit vielfältigen Objekten benötigt. Labbé u. a. haben für das

⁶Eine neuere Implementierung von MegaPose existiert auch im Rahmen der HappyPose Toolbox <https://agimus-project.github.io/happypose/>. In der vorliegenden Arbeit wurde jedoch die ursprüngliche Implementierung der Autoren verwendet.

Training ihrer Methode einen synthetischen Datensatz mit einem Umfang von 2 Millionen Bildern erzeugt. Ähnlich wie bei den synthetischen Daten der BOP-Challenge werden die Objekte zufällig aus dem Datensatz ausgewählt und mithilfe eines Physiksimulators auf eine Ebene fallen gelassen. Materialien, Hintergrundtexturen, Beleuchtung und Kamerapositionen werden dabei zufällig variiert. Der von den Autoren im Rahmen ihrer Arbeit zu MegaPose erstellte Datensatz dient dem Training vieler anderer Methoden, die fortan zur Erkennung neuartiger Objekte entwickelt werden.

Mit FoundPose [53] existiert bereits eine Weiterentwicklung durch den US-amerikanischen Konzern Meta Platforms. Örnek u. a. nutzen in FoundPose sogenannte Foundation Features (vgl. Abschnitt 2.5.3). Diese ermöglichen es der Methode, signifikant genauer und schneller als MegaPose zu sein (siehe Tabelle 3).

Methode	T-LESS	YCB-V	Zeit [s]
<i>Grobe Posenbestimmung</i>			
FoundPose	33,8	45,2	1,7
MegaPose	17,7	28,1	15,5
<i>Verfeinerung der Posenbestimmung (mit einzelner Posen-Hypothese)</i>			
FoundPose⁷	51,0	66,1	6,4
MegaPose	47,7	60,1	31,7
<i>Verfeinerung der Posenbestimmung (mit mehreren Posen-Hypothesen)</i>			
FoundPose⁸	57,0	69,0	20,5
MegaPose⁹	50,7	62,1	47,4

Tabelle 3: Vergleich von FoundPose und MegaPose auf den T-LESS und YCB-V Datensätzen, Daten aus [53]

Zum Stand vom 01.08.2024 wurde der Code für FoundPose jedoch noch nicht veröffentlicht¹⁰, weshalb die Methode im Rahmen dieser Arbeit nicht weiter untersucht werden konnte.

2.5.2. SAM6D

Die Methode SAM6D von Lin u. a. nutzt das Segment Anything Model (SAM) des US-amerikanischen Konzerns Meta [55], um die 6D-Posenbestimmung von neuartigen Objekten zu ermöglichen. Die Besonderheit von SAM6D besteht darin, dass im Vergleich zu anderen Methoden keine initiale Region of Interest für die Posenbestimmung benötigt wird. Ähnlich wie MegaPose, nutzt auch SAM6D einen mehrstufigen Ansatz, der in Abbildung 32 dargestellt ist.

¹⁰<https://evinpinar.github.io/foundpose/>

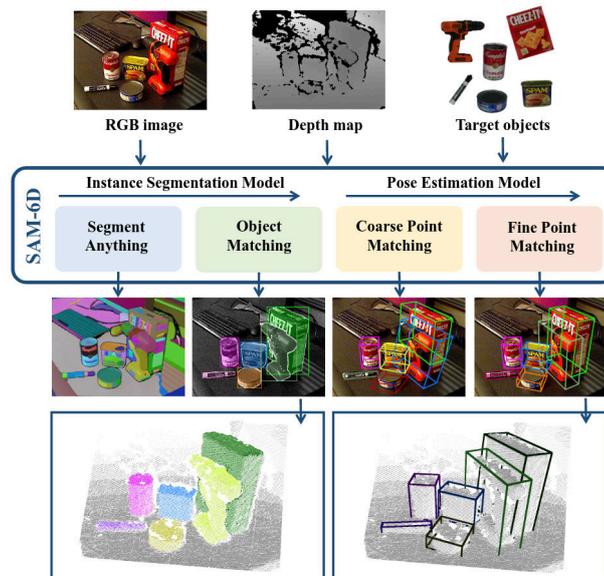


Abbildung 32: Übersicht des Ablaufs von SAM-6D [54]

SAM-6D nutzt zwei spezialisierte Submodelle, das Instance Segmentation Model (ISM) und das Pose Estimation Model (PEM). Das ISM nimmt SAM als Ausgangspunkt, um alle potenziellen Objektvorschläge zu generieren.

Das **Instance Segmentation Model** (ISM) wird unter Verwendung von SAM entwickelt, um dessen Zero-Shot-Fähigkeiten zu nutzen und klassenagnostische Vorschläge für sämtliche Objekte im Bild zu generieren. Der generelle Segmentierungsprozess von SAM ist für ein besseres Verständnis beispielhaft in Abbildung 33 dargestellt. Die von SAM generierten Vorschläge werden mit gerenderten Objektvorlagen verglichen. Die geometrische Übereinstimmung wird anhand des Intersection-over-Union (IoU)-Werts zwischen dem generierten Vorschlag und der 2D-Projektion des transformierten Objekts berechnet. Das ISM rendert zu diesem Zweck mehrere 2D-Objektmasken des zu detektierenden Objekts.

Das **Pose Estimation Model** (PEM) berechnet anschließend eine 6D-Pose für jeden identifizierten Vorschlag, der mit einem neuen Objekt übereinstimmt. Die Posenschätzung wird als Punktabgleichsproblem zwischen den abgetasteten Punktemengen des Vorschlags und des Zielobjekts formuliert, wobei Verdeckungen, Segmentierungsungenauigkeiten und Sensorausreißer berücksichtigt werden. SAM6D verwendet hierfür das PointNet, bzw. PointNet++ Modell [56, 57]. Das PEM basiert auf zwei Punktabgleichsphasen. In der ersten Phase wird eine grobe Korrespondenz ermittelt, die eine initiale Objektpose liefert und die Punktemenge des Vorschlags transformiert. In der zweiten Phase werden die Positionskodierungen beider Punktemengen integriert, um Korrespondenzen für eine präzisere Objektschätzung zu erzeugen [54].



(a) Eingabebild



(b) Ausgabebild mit segmentierten Masken

Abbildung 33: Beispiel für das Segmentieren und Erstellen von Masken mit dem SAM-Modell

Die Veröffentlichung des SAM 2 Modells im August 2024 gibt Grund zur Annahme, dass dieses Modell für zukünftige Verbesserungen im Bereich der 6D-Posenbestimmung verwendet werden könnte. SAM 2 bietet gegenüber SAM einen stärkeren Fokus auf die Segmentierung von Objekten in Videos. Durch die Verwendung von temporalen Hinweisen ist es möglich, Objekte auch dann robust und plausibel zu verfolgen, wenn diese kurzfristig den Bildbereich verlassen [58].

2.5.3. FoundationPose

Bei FoundationPose [59] handelt es sich um ein vereinheitlichtes Foundation-Modell für die 6D-Posenbestimmung und -verfolgung. Dieses Modell wurde von Mitarbeitenden des US-amerikanischen Unternehmens Nvidia Corporation entwickelt und unterstützt sowohl modellbasierte als auch modellfreie Ansätze. Der Stand der Technik am 01.08.2024 zeigt, dass FoundationPose die mit Abstand beste Methode zur Erkennung neuartiger Objekte ist. Eine Übersicht des gesamten Frameworks ist in Abbildung 34 dargestellt.

Der Begriff "modellfrei" bezeichnet in diesem Kontext die Fähigkeit von FoundationPose, die 6D-Pose eines Objekts zu bestimmen, auch wenn kein 3D-Modell des Objekts verfügbar ist. Hierzu wird ein 3D-Modell aus mehreren unterschiedlichen Frames rekonstruiert, in denen das Objekt zu sehen ist. Weitere Informationen müssen für die Posenbestimmung nicht bereitgestellt werden. Dieser Ansatz basiert auf der Methode BundleSDF von Wen u. a. [60].

Weiterhin verfügt FoundationPose über ein eigenes Object-Tracking. Nahezu alle anderen Methoden führen die Posenbestimmung ausschließlich auf Basis von einzelnen Bildframes durch. Dabei wird kein zeitlicher Zusammenhang der Frames berücksichtigt. Dies kann dazu führen, dass die

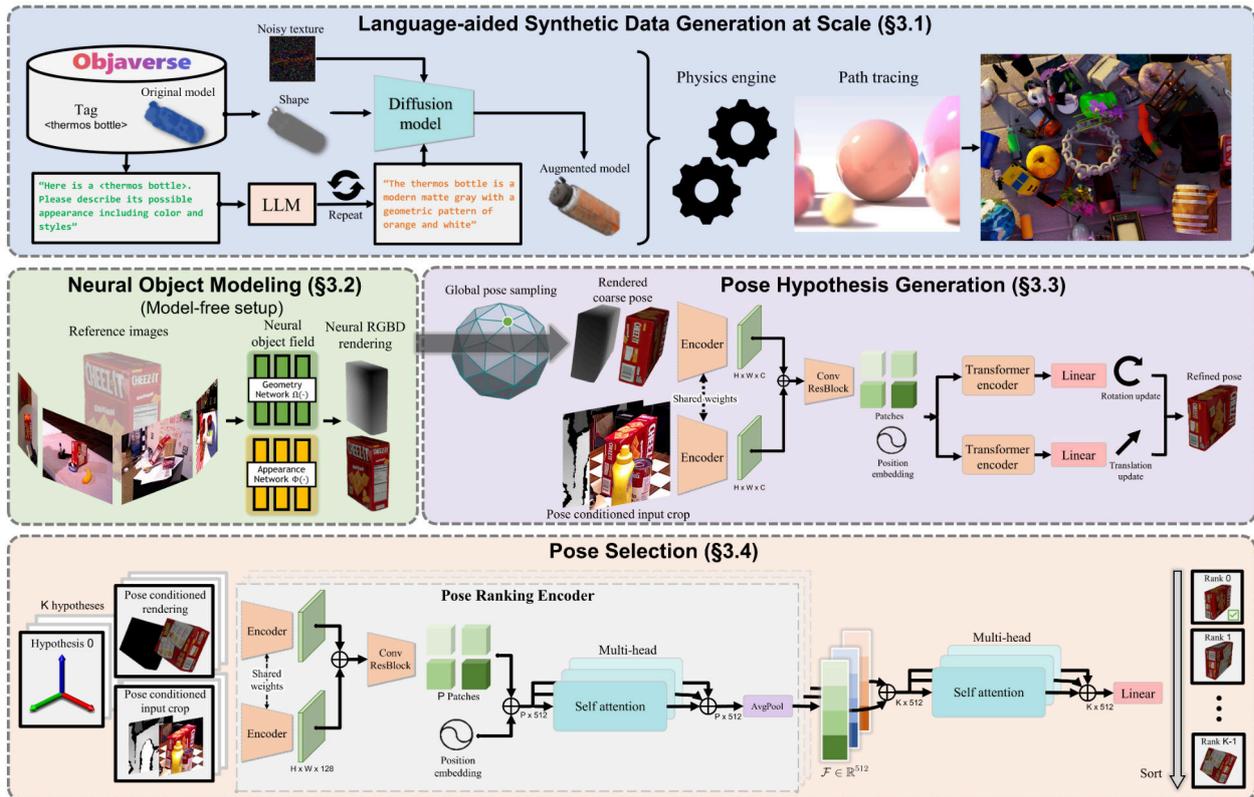


Abbildung 34: Übersicht des FoundationPose Frameworks [59]

Schätzungen der Objekte zwischen den einzelnen Frames große, nicht plausible Sprünge machen. Es ist davon auszugehen, dass zukünftige Methoden über ähnliche Funktionen verfügen werden, da robustes 6D Object-Tracking in Videos von großer Relevanz ist, jedoch von vergleichsweise wenigen Methoden adressiert wird.

Die Idee hinter einem Foundation-Modell, wie es bei FoundationPose genutzt wird, ist, dass die von diesen Modellen gelernten Repräsentationen, sogenannte Foundation Features, allgemeingültig genug sind, um eine Vielzahl von Aufgaben in unterschiedlichen Domänen zu unterstützen, ohne dass von Grund auf neu trainierte Modelle benötigt werden. Foundation Features müssen aus besonders umfassenden und vielfältigen Datensätzen extrahiert werden. Zur Erstellung eines solchen Datensatzes haben Wen u. a. eine neuartige Pipeline zur Generierung synthetischer Daten für das Training entwickelt, die auf den neuesten Ressourcen und Techniken basiert.

Objaverse liefert eine umfangreiche 3D-Modell-Datenbank, während Large Language Models und Diffusionsmodelle die Augmentierung von Daten ermöglichen, siehe Abbildung 35. Dieser Ansatz erhöht sowohl die Menge als auch die Vielfalt der Daten im Vergleich zu früheren Arbeiten wie MegaPose erheblich [59].

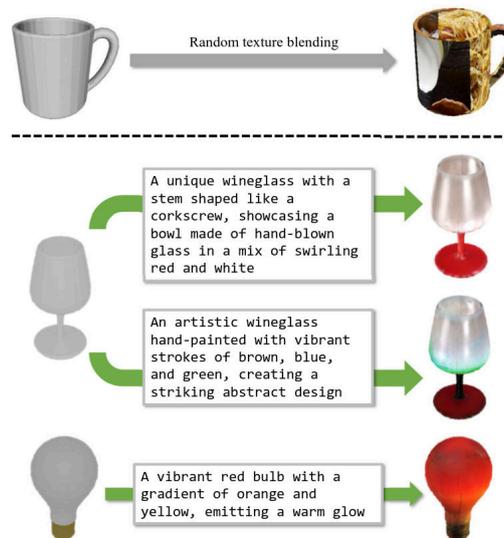


Abbildung 35: Vergleich zwischen zufälliger Texturüberblendung und der von FoundationPose verwendeten LLM-unterstützten Methode [59]

Aus den Foundation Features lassen sich anschließend sogenannte Attention Layer aufbauen. Hierbei handelt es sich um einen Mechanismus des maschinellen Lernens, der es einem Modell ermöglicht, dynamisch den Fokus auf unterschiedliche Teile einer Eingabesequenz zu legen, je nach dem Kontext und der Aufgabe, die gelöst werden soll. Dieses Prinzip hat aktuellen Entwicklungen im Bereich der Large Language Models zum Erfolg verholfen, vgl. [61]. Im Falle von FoundationPose ermöglichen die Foundation Features eine besonders schnelle und robuste 6D-Posenbestimmung.

FoundationPose wird voraussichtlich weiterentwickelt und als fertiges Produkt in das NVIDIA Omniverse¹¹ integriert werden.

¹¹<https://www.nvidia.com/de-de/omniverse/>

2.6. HRC-HUB - ein holistisches Framework für die MRK

Im Rahmen dieser Arbeit wurde ein Hand-Interaktionsmodell für eine durch Augmented Reality unterstützte Mensch-Roboter-Kollaboration entwickelt, welches als Beitrag in den 73. CIRP Annals - Manufacturing Technology veröffentlicht wurde [3].

Um kleinen und mittleren Unternehmen (KMU) bei ihren Automatisierungsherausforderungen zu unterstützen, wurde der HRC-Hub entwickelt. Dieses Framework erfüllt viele der Anforderungen, die bei der Integration neuer MRK-Arbeitsplätze auftreten. Der HRC-Hub ermöglicht es Unternehmen, Arbeitsplätze schnell und effizient mit Augmented Reality anzupassen und fördert darüber hinaus eine herstellerunabhängige, menschenzentrierte Automatisierung, siehe Abbildung 36.

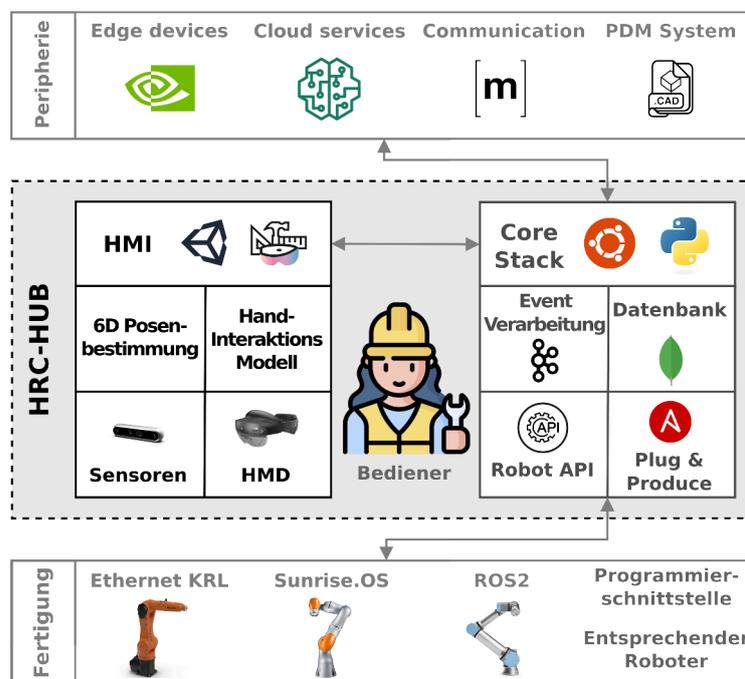


Abbildung 36: Übersicht der Architektur des HRC-Hubs

Die Fertigungsebene umfasst kollaborative Roboter, während eine weitere Ebene für die Vernetzung peripherer Geräte und Datenbanken zuständig ist. Der Bediener im HRC-HUB stellt die Schnittstellen zu den technischen Systemen dar. Die Verwendung universeller Programmierschnittstellen sowie middleware-basierter Ansätze erlaubt eine flexible Erweiterbarkeit des Systems.

Obwohl der Average Recall der in Abschnitt 2.5 vorgestellten Methoden für die Posenbestimmung statischer Objekte ausreichend sein kann, gewährleistet er keine genaue Erkennung bewegter Objekte. Deshalb wurde ein innovatives Hand-Interaktionsmodell entwickelt, siehe Abbildung 37.

Zustand	aus der Hand		Kontakt	in der Hand		Kontakt
Objekt-kollider	 Box 8 cm	 Mesh		 Box 1.5 cm		
Vorgang	untätig	Annäherung	Greifen	Bewegung	Positionieren	Loslassen
Virtueller Indikator						

Abbildung 37: Funktionsweise des Hand-Interaktionsmodells

Dieses Modell ist weniger anfällig für Verdeckungen und erlaubt die systematische Zerlegung von Montageaufgaben in grundlegende Operationen. Die Umsetzung erfolgt durch die Kombination von Hand-Tracking während der Demonstrationsphase und der Nutzung verschiedener Objekt-Kollider.

Die Objektnachverfolgung stellt einen essenziellen Aspekt während der Montage-Demonstration im Rahmen des Programmierens durch Vormachen dar. Dieser Prozess lässt sich in vier Phasen unterteilen, wobei für jede Phase unterschiedliche Methoden zur Posenbestimmung existieren, siehe Abbildung 38.

Stufe der Posenbestimmung	Posenbestimmung für das Tracking von Objekten				 Der Mensch in der Schleife: Korrektur der beobachteten Abweichung zwischen realem Teil und virtuellem Objekt
Initiale ROI	Festes Werkstück Magazin	Marker-basierter Bereich	Objekt-erkennung mit festem Sensor	Objekt-erkennung mit HMD	
Initiale Pose	fester Sensor trainierte Objekte	Fester Sensor neuartige Objekte	HMD trainierte Objekte	HMD neuartige Objekte	
Bewegtes Objekte	Gebunden an Hand-Interaktionsmodell		Konitnuierliche Posenbestimmung		
Zielpose	Geometrische Randbedingungen	Zusätzliche Posenbestimmung	Abgeleitet aus Hand-Tracking		
robust		flexibel			

Abbildung 38: Unterschiedliche Methoden zur Posenbestimmung für das Tracking von Objekten

Diese Phasen umfassen die Bestimmung der Region of Interest, die exakte Objektpose zu Beginn der Demonstration sowie die Posenbestimmung während der Bewegung und die Endposition. Im weiteren Verlauf der Arbeit sollen weitere Erkenntnisse über die Eignung aktueller Methoden für den industriellen Einsatz in den unterschiedlichen Stufen der Posenbestimmung gewonnen werden. Weiterhin soll perspektivisch eine methodische Bewertung der erreichbaren Genauigkeit des Hand-Interaktionsmodells ermöglicht werden

3. Zielsetzung der Arbeit

In diesem Kapitel wird, basierend auf den zuvor dargelegten Grundlagen und dem aktuellen Stand der Technik, die Zielsetzung dieser Arbeit nochmals zusammengefasst, siehe Abbildung 39.

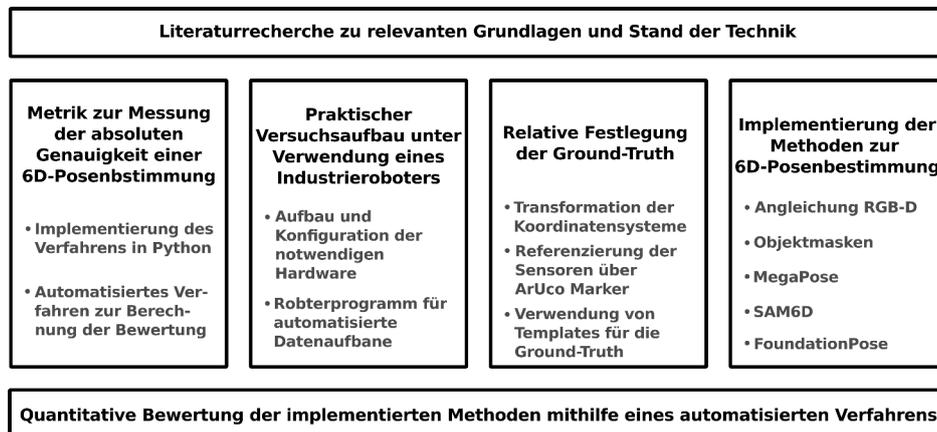


Abbildung 39: Schematische Darstellung der Arbeitsziele

Die eingeschränkte Verfügbarkeit von Datensätzen stellt eine signifikante Herausforderung in vielen Forschungs- und Anwendungsbereichen dar. Diese Einschränkung ist größtenteils auf den hohen Aufwand zurückzuführen, der mit der Annotation solcher Daten verbunden ist. Obwohl Fortschritte in der synthetischen Erzeugung von Trainingsdatensätzen verzeichnet werden, bleibt der Bedarf an realen und korrekt annotierten Testdaten bestehen, um valide Aussagen über die Güte von Methoden zur Posenbestimmung treffen zu können.

Diese Arbeit zielt einerseits darauf ab, Lösungen zu entwickeln und zu evaluieren, die den Prozess der Datensatzerstellung und -annotation effizienter gestalten.

Andererseits erlauben die derzeit verwendeten Metriken zur Bewertung einer 6D-Posenbestimmung nur bedingt Aussagen darüber, ob diese für einen industriellen Einsatz geeignet sind. Es soll daher eine methodische Bewertung der absoluten Genauigkeit einer 6D-Posenbestimmung erarbeitet werden. Unter Verwendung dieser Metrik sollen implementierte State-of-the-Art-Methoden zur 6D-Posenbestimmung unter Verwendung der automatisierten Testanlage bewertet werden.

Insgesamt soll diese Arbeit ein Framework zur Aufnahme neuer realer (Trainings-)Datensätze liefern sowie eine quantitative Bewertung von Methoden ermöglichen und somit eine solide Grundlage für vertiefende experimentelle Untersuchungen schaffen.

4. Methodik und konzeptionelle Gestaltung

Das vorliegende Kapitel dient der Erläuterung der Methodik und konzeptionellen Gestaltung der vorliegenden Arbeit. Zu Beginn erfolgt eine Darlegung der relevanten Grundlagen der Bildverarbeitung. Im Anschluss wird eine detaillierte Darstellung der Berechnung der Genauigkeit nach der Norm EN ISO 9283 präsentiert. Daraufhin erfolgt eine theoretische Analyse der Genauigkeit der eingesetzten Sensoren. Abschließend werden das zugrunde liegende Konzept sowie die Planung der durchgeführten Versuche vorgestellt.

4.1. Bilderverarbeitung

Die 6D-Posenbestimmung basiert zu großen Teilen auf Methoden und Verfahren, die der Computer Vision bzw. industriellen Bildverarbeitung zuzuordnen sind. Die hierfür relevanten Methoden werden im Folgenden kurz vorgestellt.

Das Ziel der 6D-Posenbestimmung ist es, die Lage und Orientierung eines Objektes in der Welt zu schätzen. Die Umrechnung eines Punktes $p = (u, v)$, welcher mit der Kamera aufgenommen wird, zu dem korrespondierenden Punkt $P_w = (X_w, Y_w, Z_w)$ im Weltkoordinatensystem erfolgt mit Hilfe des Lochkamera-Modells, siehe Abbildung 40.

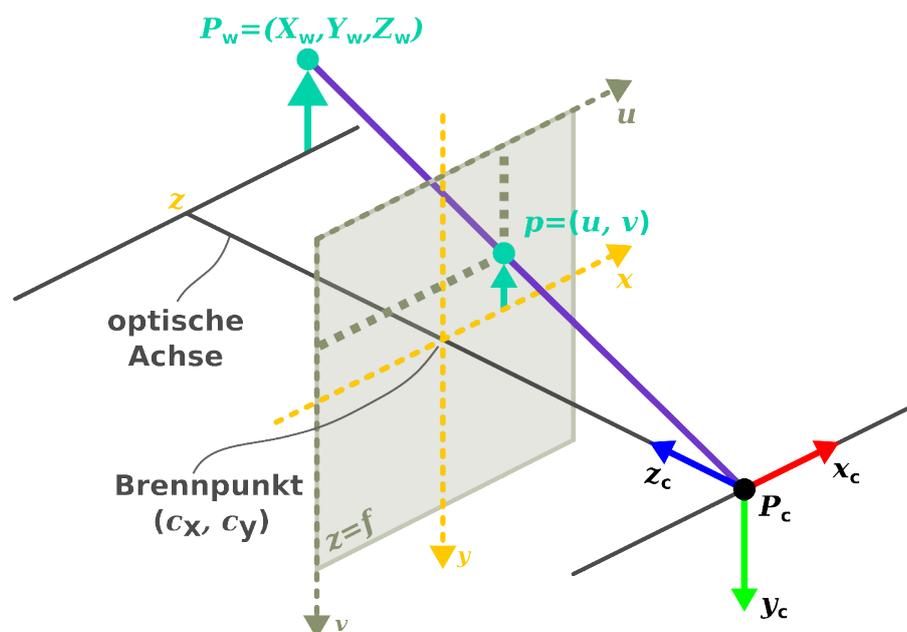


Abbildung 40: Schematische Darstellung des Lochkamera-Modells, adaptiert aus [62]

Die ausführliche Umrechnung mit den erforderlichen Zwischenschritten erfolgt nach [62] wie folgt¹². Der generelle Zusammenhang zwischen dem Bildpunkt p und dem korrespondierenden Punkt im Weltkoordinatensystem P_w bzw. im Kamerakoordinatensystem P_c lautet:

$$p = K \begin{bmatrix} R|t \end{bmatrix} P_w \quad (4.1)$$

$$p = KP_c \quad (4.2)$$

Dabei ist K die Kameramatrix, welche sich wie folgt aus den intrinsischen Parametern der Kamera zusammensetzt:

$$K = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \quad (4.3)$$

Hierbei sind c_x und c_y die Koordinaten des *Brennpunktes* in der Bildebene und f_x sowie f_y sind die Brennweiten der Kamera. Im Falle, dass der Brennpunkt in der Mitte des Bildes liegt, gilt für die Brennweite $f_x = f_y$. Bei kalibrierten Sensoren, wie sie im Rahmen dieser Arbeit verwendet wurden (vgl. Abschnitt 2.4.3), lassen sich die notwendigen Parameter direkt vom Sensor ausgehen. Ist dies nicht der Fall, ist zunächst eine manuelle Kalibrierung des Sensors notwendig.

Unter Verwendung von K ergibt sich nun die folgende Gleichung für die Umrechnung von Punkt p mit Koordinaten (u, v) und der Lage des Punktes P_c im Koordinatensystem der Kamera:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} \quad (4.4)$$

Mit Hilfe der homogenen Transformation $\begin{bmatrix} R|t \end{bmatrix}$:

$$\begin{bmatrix} R|t \end{bmatrix} = \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.5)$$

¹²Anmerkung: Die Berechnungen vernachlässigen den Skalierungsfaktor s , da dieser nicht Bestandteil des Lochkamera-Modells ist. (s kann zur Umrechnung bei unterschiedlichen Bildauflösungen verwendet werden.)

kann anschließend die Umrechnung von P_w zu P_c erfolgen:

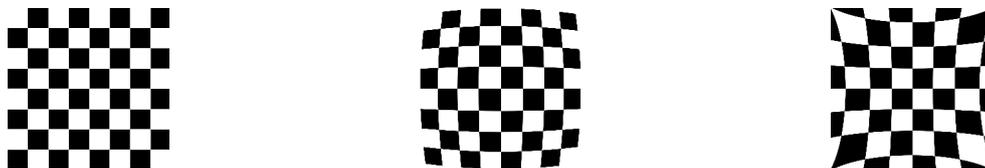
$$P_c = \begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \quad (4.6)$$

Abschließend lässt sich der folgende Zusammenhang zwischen dem Bildpunkt p , der Kameramatrix K , der homogenen Transformation $\begin{bmatrix} R|t \end{bmatrix}$ und dem Punkt im Weltkoordinatensystem P_w aufstellen:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \quad (4.7)$$

Dieser Zusammenhang ermöglicht es, mit dem Lochkammermodell die Verbindung zwischen 3D-Punkten in der realen Welt und ihren 2D-Projektionen im Kamerabild herzustellen. Für eine ausführlichere Erklärung, die auch andere Kameramodelle und Multi-View-Ansätze behandelt, sei auf [63] verwiesen.

Ein weiterer wichtiger Aspekt ist der Umgang mit einer möglichen Verzeichnung der aufgenommenen Bilder, siehe Abbildung 41. Die Verzeichnung, oft fälschlicherweise als optische Verzerrung bezeichnet, ist ein geometrischer Abbildungsfehler optischer Systeme, der zu einer lokalen Veränderung des Abbildungsmaßstabes führt (vgl. [64]).



(a) verzeichnungsfreie Abbildung (b) tonnenförmige Verzeichnung (c) kissenförmige Verzeichnung

Abbildung 41: Effekt der Verzeichnung am Beispiel eines Schachbrettmusters, in Anlehnung an [62]

Die im Rahmen dieser Arbeit verwendeten Sensoren ermöglichen die Aufnahme verzerrungsfreier Bilder. Ist dies nicht der Fall, müssen die Verzerrungsparameter der jeweiligen Kamera zunächst ermittelt werden. Hierzu müssen Kalibrierungsverfahren genutzt werden, analog zur Ermittlung der Kameramatrix. Anschließend kann eine Korrektur der Bilder vorgenommen werden.

4.2. Berechnung der Genauigkeit

Wie in Kapitel 2.4.1 beschrieben, existieren verschiedene Metriken zur Bewertung der erreichten Genauigkeit einer Posenbestimmung. Diese Metriken wurden jedoch in erster Linie entwickelt, um einen möglichst objektiven Vergleich von Methoden zur Posenbestimmung untereinander zu ermöglichen. Daher liegt der primäre Fokus dieser Metriken nicht darauf, zu bewerten, inwiefern eine Methode für einen spezifischen Anwendungsfall geeignet ist.

Um beurteilen zu können, ob eine Posenbestimmung den Anforderungen eines bestimmten Prozesses genügt, ist es notwendig, absolute Kennwerte zu ermitteln. Im Rahmen dieser Arbeit soll daher eine Metrik verwendet werden, die an die EN ISO 9283 angelehnt ist [65]. Diese Norm beschreibt „Leistungskenngrößen und zugehörige Prüfmethode“ für Industrieroboter. Die in dieser Norm definierten Kennwerte sollen im weiteren Verlauf der Arbeit genutzt werden, um die Genauigkeit der implementierten Methoden zur 6D-Posenbestimmung zu bewerten.

Die DIN EN ISO 9283 unterscheidet zwischen (Absolut-)Genauigkeit und Wiederholgenauigkeit. Zudem differenziert die Norm zwischen der Genauigkeit in Bezug auf die Position und die Orientierung.

Für die Position ergeben sich somit die *Positionsgenauigkeit* AP_p sowie die *Positionswiederholgenauigkeit* RP_p . Abbildung 42 stellt diese graphisch dar.

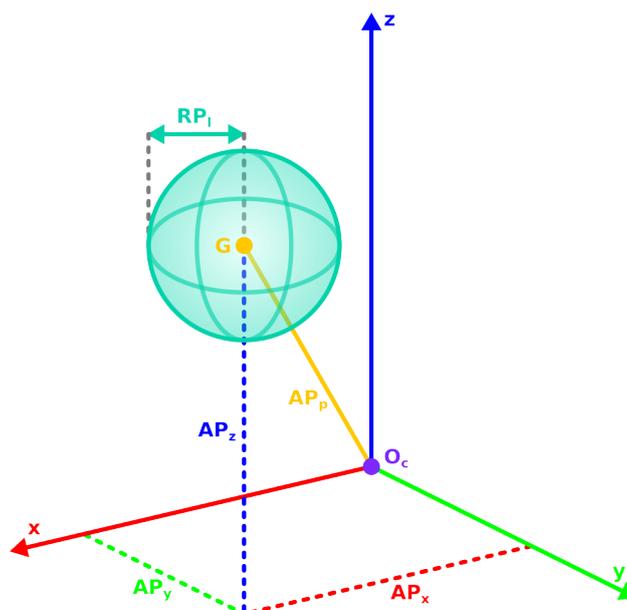


Abbildung 42: Positionsgenauigkeit und -Wiederholgenauigkeit nach DIN EN ISO 9283 [65]

Die *Positionsgenauigkeit* AP_p beschreibt die Differenz zwischen der Position einer Sollpose und dem Schwerpunkt der Istpositionen:

$$AP_p = \sqrt{AP_x^2 + AP_y^2 + AP_z^2} \quad (4.8)$$

mit:

$$AP_x = (\bar{x} - x_c) \quad (4.9)$$

$$AP_y = (\bar{y} - y_c) \quad (4.10)$$

$$AP_z = (\bar{z} - z_c) \quad (4.11)$$

Hierbei sind \bar{x} , \bar{y} , \bar{z} die Koordinaten des Schwerpunktes der Gruppe von Punkten, die nach n -facher Wiederholung derselben Pose erreicht werden:

$$\bar{x} = \frac{1}{n} \sum_{j=1}^n x_j \quad (4.12)$$

$$\bar{y} = \frac{1}{n} \sum_{j=1}^n y_j \quad (4.13)$$

$$\bar{z} = \frac{1}{n} \sum_{j=1}^n z_j \quad (4.14)$$

Die *Positionswiederholungsgenauigkeit* RP_l gibt die Genauigkeit der Übereinstimmung zwischen den Ist-Positionen nach n wiederholten Anläufen zur selben Soll-Position in derselben Richtung an:

$$RP_l = \bar{l} + 3S_l \quad (4.15)$$

Dabei ist:

$$\bar{l} = \frac{1}{n} \sum_{j=1}^n l_j \quad (4.16)$$

$$l_j = \sqrt{(x_j - \bar{x})^2 + (y_j - \bar{y})^2 + (z_j - \bar{z})^2} \quad (4.17)$$

Sowie S_l die Standardabweichung:

$$S_l = \sqrt{\frac{\sum_{j=1}^n (l_j - \bar{l})^2}{n - 1}} \quad (4.18)$$

Analog zur Positionsgenauigkeit werden die Genauigkeit der Orientierung (*Orientierungsgenauigkeit*) sowie die Wiederholgenauigkeit der Orientierung (*Orientierungswiederholgenauigkeit*) berechnet, siehe Abbildung 43.

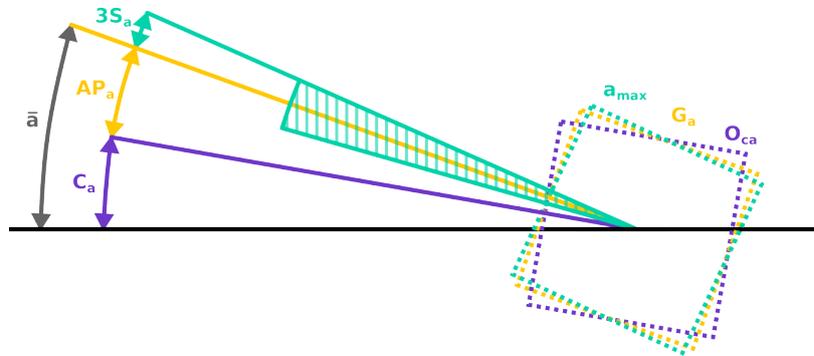


Abbildung 43: Orientierungsgenauigkeit und -Wiederholgenauigkeit für a nach DIN EN ISO 9283 [65]

Dabei ist C_a die Sollorientierung in Bezug auf das entsprechende Koordinatensystem, und \bar{a} steht für den Mittelwert der Istorientierungen. Die Orientierungsgenauigkeit AP_a wird als Differenz zwischen der Orientierung einer Sollposition und dem Mittelwert der Istorientierungen definiert. Analog dazu gilt dies auch für die Orientierungen b und c :

$$\bar{a} = \frac{1}{n} \sum a_j \quad (4.19)$$

$$\bar{b} = \frac{1}{n} \sum b_j \quad (4.20)$$

$$\bar{c} = \frac{1}{n} \sum c_j \quad (4.21)$$

Für die Orientierungs-Wiederholgenauigkeiten gilt:

$$RP_a = \pm 3S_a = \pm 3 \sqrt{\frac{\sum_{j=1}^n (a_j - \bar{a})^2}{n-1}} \quad (4.22)$$

$$RP_b = \pm 3S_b = \pm 3 \sqrt{\frac{\sum_{j=1}^n (b_j - \bar{b})^2}{n-1}} \quad (4.23)$$

$$RP_c = \pm 3S_c = \pm 3 \sqrt{\frac{\sum_{j=1}^n (c_j - \bar{c})^2}{n-1}} \quad (4.24)$$

Wobei S_a , S_b und S_c die Standardabweichungen sind.

Die Bewertung kann auf die 6D-Posenbestimmung übertragen werden, indem die Genauigkeit als Differenz zwischen Posenschätzung und Ground-Truth für eine mehrfache Posenbestimmung angegeben wird, siehe Abbildung 44.

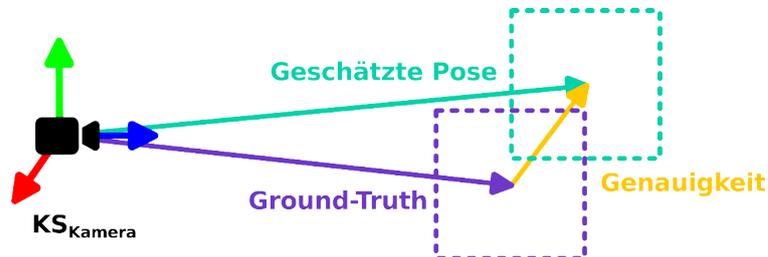


Abbildung 44: Genauigkeit als Differenz zwischen geschätzter Pose und Ground-Truth

Anschließend erfolgt die Implementierung der Metriken in Python. Für die Pose-Wiederholgenauigkeit RP_l wird die folgende Unterfunktion für die Berechnung verwendet:

```

317 def calc_RP_l(x, y, z, ap_x, ap_y, ap_z) -> float:
318     l = []
319
320     for i in range(len(x)):
321         error = np.sqrt((x[i] - ap_x) ** 2 + (y[i] - ap_y) ** 2 + (z[i] - ap_z) ** 2)
322         l.append(error)
323
324     l_mean = np.mean(l)
325
326     S_l = np.sqrt(np.sum([(lj - l_mean) ** 2 for lj in l]) / (len(l) - 1))
327
328     RP_l = l_mean + 3 * S_l
329
330     return RP_l

```

Quellcode 1: Implementierung zur Berechnung der Pose-Wiederholgenauigkeit

Die Implementierung der Orientierungs-Wiederholgenauigkeiten RP_a , RP_b und RP_c lassen sich analog wie folgt bestimmen:

```

333 def calc_RP_abc(a, b, c, ap_a, ap_b, ap_c) -> list:
334     S_a = np.sqrt(np.sum([(ai - ap_a) ** 2 for ai in a]) / (len(a) - 1))
335     RP_a = [S_a * 3, -1 * S_a * 3]
336
337     S_b = np.sqrt(np.sum([(bi - ap_b) ** 2 for bi in b]) / (len(b) - 1))
338     RP_b = [S_b * 3, -1 * S_b * 3]
339
340     S_c = np.sqrt(np.sum([(ci - ap_c) ** 2 for ci in c]) / (len(c) - 1))
341     RP_c = [S_c * 3, -1 * S_c * 3]
342
343     return RP_a, RP_b, RP_c

```

Quellcode 2: Implementierung zur Berechnung der Orientierungs-Wiederholgenauigkeiten

Im Rahmen der praktischen Implementierung werden die zuvor erläuterten Metriken eingesetzt, um die Genauigkeiten der implementierten Methoden zur 6D-Posenbestimmung zu evaluieren.

4.3. Theoretische Genauigkeit der Sensoren

Gemäß der goldenen Regel der Messtechnik, welche besagt, dass die Unsicherheit eines Messgeräts ein Zehntel, im äußersten Fall ein Fünftel der Toleranz nicht überschreiten sollte [66], muss vor der methodischen Bewertung der Genauigkeit einer 6D-Posenbestimmung zunächst sichergestellt werden, dass das Messmittel, in diesem Fall die verwendeten RGB-D Sensoren, eine ausreichende Genauigkeit besitzt, um verlässliche Aussagen zu ermöglichen. In diesem Kapitel erfolgt eine kurze Untersuchung der theoretisch erreichbaren Genauigkeit unter Verwendung der zuvor vorgestellten Sensoren.

Der Zusammenhang zwischen der Auflösung eines Sensors und der theoretischen Genauigkeit, die dieser abbilden kann, lässt sich durch die relative Pixelgröße auf dem Abbildungsobjekt beschreiben. Diese entspricht dem Bereich des Objekts, der von einem einzelnen Pixel des Sensors erfasst wird.

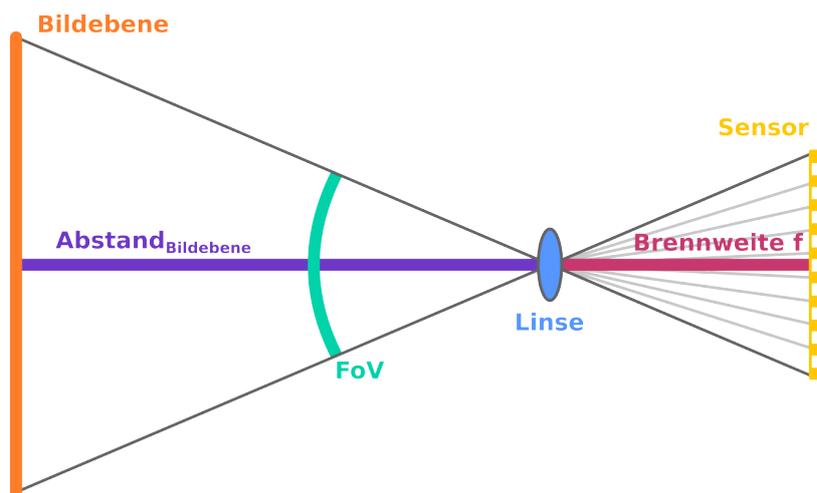


Abbildung 45: Schematisches Kameramodell zur Auflösung der Bildebene

Wie zu erkennen ist, hängt die relative Pixelgröße von mehreren Faktoren ab. Dazu zählen die Anzahl der Pixel des Sensors, die das kleinste auflösbare Element aus Sensor-Seite darstellen. Auf diese Anzahl wird die Bildebene projiziert. Der Umfang der Bildebene hängt dabei sowohl vom Abstand zur Bildebene als auch vom Field of View des Sensors ab. Das Field of View ist dabei umgekehrt proportional zur Brennweite f .

Der dargestellte Zusammenhang lässt sich mathematisch wie folgt ausdrücken:

$$\text{Relative Pixelgröße} \left[\frac{mm}{px} \right] = \frac{\sin(\text{FoV}_{\text{Sensor}}^\circ) * \text{Abstand}_{\text{Bildebene}} [mm]}{\text{Auflösung}_{\text{Sensor}} [px]} \quad (4.25)$$

Abbildung 46 stellt die relative Pixelgröße eines Sensors mit einem horizontalen Field of View von 69° graphisch dar. Dabei werden übliche horizontale Auflösungen verwendet und ein maximaler Abstand von 5 m zu Bildebene verwendet.

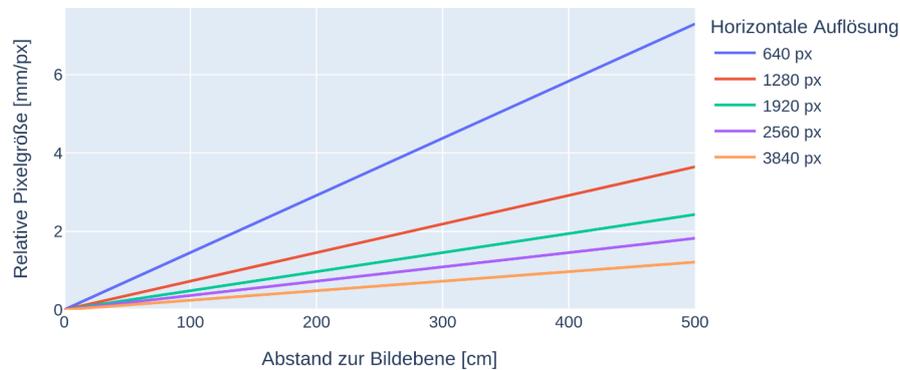


Abbildung 46: Relative Pixelgröße in Abhängigkeit der Auflösung (bei FoV 69°)

Wie zu erkennen ist, kann es je nach Auflösung des Sensors und Abstand zur Bildebene zu erheblichen Einschränkungen in Bezug auf die theoretisch abbildbare Genauigkeit des Sensors kommen. Bei einem Abstand von 4 m kann ein Sensor mit einer horizontalen Auflösung von 640 Pixeln etwa nur noch circa 6 mm pro Pixel abbilden.

Die Abhängigkeit des Field of Views bei einer horizontalen Auflösung von 1280 Pixeln wird in Abbildung 47 dargestellt.

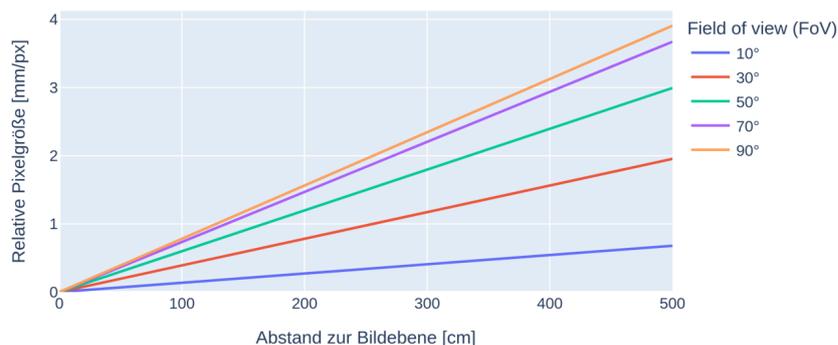
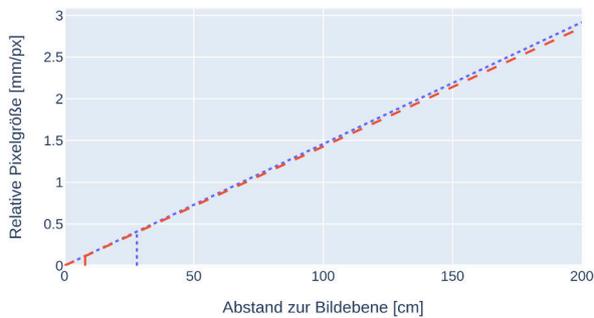


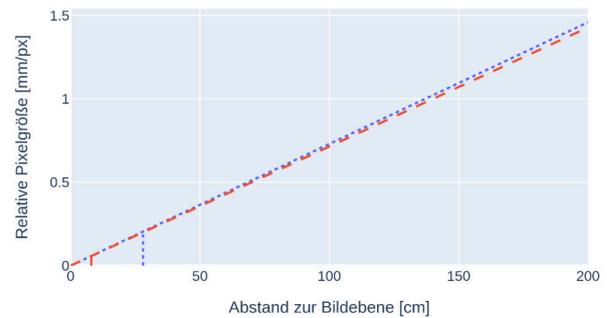
Abbildung 47: Relative Pixelgröße in Abhängigkeit des FoV (bei 1280 px)

Es lässt sich feststellen, dass ein kleines Field of View prinzipiell eine höhere theoretische Genauigkeit ermöglicht. Allerdings ist die wahrnehmbare Fläche in der Bildebene dadurch erheblich kleiner, da dieser Zusammenhang dem klassischen *Zoom*-Effekt in der Fotografie entspricht. In entsprechenden Fällen stellt die Verwendung von Objektiven mit einer passenden Brennweite jedoch eine Möglichkeit dar, um die darstellbare relative Pixelgröße zu optimieren.

Anhand der in Tabelle 2 dargestellten Daten können die Werte für die horizontale relative Pixelgröße der Sensoren berechnet werden, die in Abbildung 48 graphisch dargestellt sind. Diese Berechnungen wurden für die beiden verwendeten Auflösungen von 1280x720 Pixeln (720p) und 640x360 Pixeln (360p) durchgeführt.



(a) 640 px



(b) 1280 px

Abbildung 48: Relative Pixelgröße der verwendeten Sensoren je nach horizontaler Auflösung: **RealSense D435i (gepunktet)** und **Luxonis OAK-D Pro (gestrichelt)**, die vertikale Linie stellt die jeweils geringste wahrnehmbare Tiefe dar

Aus der Darstellung lässt sich eine theoretische Genauigkeit von circa 2,2 mm für eine Auflösung von 480p sowie circa 1,1 mm für 720p bei einem Abstand von jeweils 150 cm ableiten. Dieser Abstand entspricht in etwa dem maximalen Abstand, der mit dem praktischen Versuchsaufbau theoretisch erreicht werden kann. Gemäß der zu Beginn dieses Kapitels vorgestellten Regel würde dies implizieren, dass im Falle einer Auflösung von 480p nur eine Genauigkeit von 11-22 mm gemessen werden kann. Da die Ground Truth jedoch um ein Vielfaches genauer ist und etwaige Messfehler eines Sensors ebenfalls Bestandteil einer 6D-Posenbestimmung sind, ist dies akzeptabel. Sollte eine Methode jedoch innerhalb dieser theoretischen Genauigkeit liegen, ist anzunehmen, dass diese folglich durch die Auflösung des Sensors limitiert ist.

Weiterhin liegt die Erkenntnis nahe, dass eine höhere Sensoraufklärung eine höhere Genauigkeit ermöglicht. Dabei ist jedoch zu beachten, dass mit steigender Auflösung auch der Rechenaufwand für die Methoden zunimmt (vgl. Abschnitt 5.3). Für höhere Auflösungen muss daher auf Hardware aus dem Enterprise-Segment zurückgegriffen werden, was zu deutlich höheren Kosten führt. Zudem wird auch die benötigte Zeit für die Posenbestimmung zunehmen.

4.4. Konzept und Versuchsplanung

Im Folgenden wird das Konzept des Testaufbaus kurz erläutert und auf den Umfang und Aufbau der Versuchsreihen eingegangen. Dabei wird starker Bezug auf die Zielsetzung der Arbeit in Kapitel 3 genommen.

Zur Erprobung und anschließenden Validierung des entwickelten Konzepts wurde eine kleine Auswahl an Objekten aus bestehenden Datensätzen getroffen (vgl. Abschnitt 2.4.2). Somit ist einerseits sichergestellt, dass 3D-Modelle der Objekte zur Verfügung stehen. Weiterhin können die Ergebnisse der Evaluierung qualitativ mit den bestehenden Werten der BOP-Challenge verglichen werden. Eine tabellarische Übersicht der verwendeten Objekte findet sich in Tabelle 4.

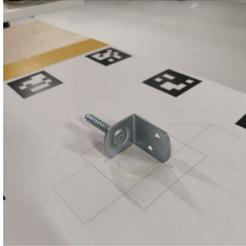
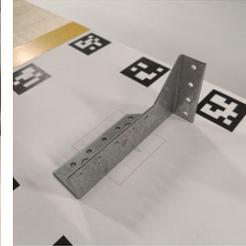
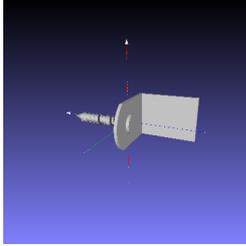
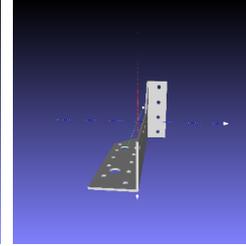
Objekt ID	000002	000006	000014	000021
Datensatz	HOPE	ITODD	HOPE	ITODD
Beschreibung	BBQ Soße	Zaunhalter	Milchkarton	Balkenschuh
Foto				
CAD-Modell				

Tabelle 4: Übersicht der verwendeten Testobjekte

Die Auswahl umfasst sowohl texturierte als auch nicht texturierte Objekte und berücksichtigt verschiedene Objektdimensionen. Verwendet wurden texturierte Objekte aus dem HOPE-Datensatz und texturlose Objekte aus dem ITODD-Datensatz, da reale Modelle dieser Objekte verfügbar sind.

Bei der Auswahl der Objekte war es zudem wichtig, dass diese keine Symmetrien aufweisen, da diese nicht von der Metrik zur Evaluierung berücksichtigt werden. Bei den Objekten 000002 und 000014 weist die Geometrie zwar Symmetrien auf, jedoch sorgt die Textur der Objekte dafür, dass diese nicht symmetrisch sind.

Zur automatisierten Datenaufnahme wird ein MRK-fähiger Industrieroboter verwendet, der LBR iiwa 14 R820 des Herstellers KUKA. Dieser verfügt über eine Positionsgenauigkeit von $\pm 0,15$ mm [67] und ist somit um ein Vielfaches genauer als die theoretische Genauigkeit der verwendeten Sensoren (vgl. Abschnitt 4.3). Abbildung 49 zeigt eine Abbildung des verwendeten Roboters im Aufbau sowie die drei Bahnen, die dieser für die Datenaufnahme fährt.

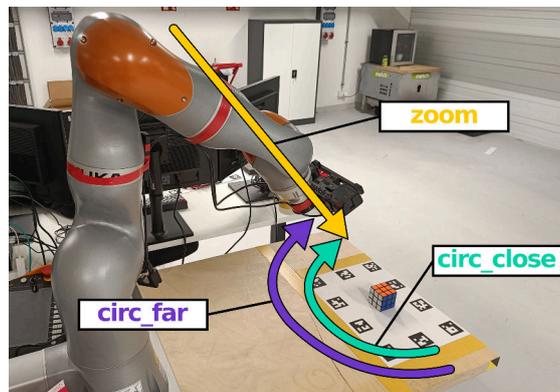


Abbildung 49: Exemplarische Darstellung der drei Roboterbewegungen zur Datenaufnahme

Die Kreisbahnen für *circ_close* und *circ_far* simulieren dabei aus unterschiedlichem Abstand die Bewegung, die typisch für eine manuelle Handhabung der Objekte ist, wobei der Betrachtungswinkel in etwa dem eines Head-Mounted Displays entspricht. Die lineare Bahn *zoom* bietet im Vergleich dazu unterschiedliche Abstände zum Objekt bei ansonsten unveränderter Orientierung.

Zusätzlich zu den drei unterschiedlichen Bahnen sollen verschiedene Grade an Objektverdeckung durch die Hand eines Benutzers abgebildet werden, siehe Abbildung 50.

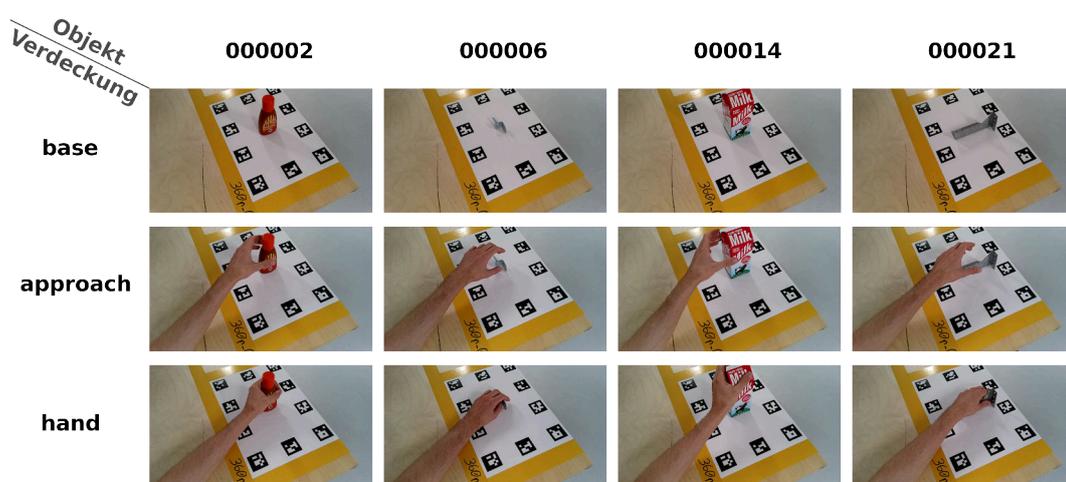


Abbildung 50: Übersicht der verschiedenen Arten der Objektverdeckung durch Hand-Objekt-Interaktionen für die Versuchsreihen

Aus dem unterschiedlichen Grad an Verdeckung ergeben sich die drei Versuchsreihen *base* ohne Verdeckung, *approach* mit partieller Verdeckung, die der Annäherung kurz vor dem Greifen entspricht, sowie *hand* mit der maximalen Verdeckung durch die Hand des Bedieners. Dies soll erste Erkenntnisse in Bezug auf die Eignung einer 6D-Posenbestimmung während einer Hand-Objekt-Interaktion ermöglichen. Es ist zu beachten, dass dieser Vergleich rein qualitativ ist, da der proportionale Anteil an Verdeckung je nach Objektgröße variiert. Mehr hierzu in Kapitel 7.

Weiterhin sollen die Auflösung der Sensoren, die Bildfrequenz (Frames per Second, FPS), mit der die Testdaten aufgenommen werden, sowie die Geschwindigkeit (Velocity, VEL), mit der der Roboter die Bahnen fährt, variiert werden. Die Variation der Bildrate soll Aufschluss darüber geben, inwiefern das Objekt-Tracking der FoundationPose-Methode gegenüber größeren Abständen zwischen den einzelnen Frames robust ist. Durch die Variation der Geschwindigkeit soll untersucht werden, ob unscharfe Frames entstehen können und welche Auswirkungen dies auf die Posenbestimmung hat.

Zusammengefasst ergeben sich acht Versuche. Diese sind in Tabelle 5 tabellarisch dargestellt, wobei alle wichtigen Parameter angegeben sind¹³.

Versuch	Auflösung	FPS	VEL	Verdeckung			Methode		
				base	approach	hand	MegaPose	SAM6D	FoundationPose
A1	360p	6	100	x	x	x	x	x	x
A2	720p	6	100	x	x	x	x	x	x
B1	720p	30	100	x					x
B2	720p	15	100	x					x
B3	720p	6	100	x					x
C1	720p	6	100	x					x
C2	720p	6	60	x					x
C3	720p	6	20	x					x

Tabelle 5: Übersicht der Versuchsreihen

Basierend auf dem entwickelten Konzept soll im weiteren Verlauf der Arbeit zunächst die praktische Implementierung beschrieben werden, bevor die Ergebnisse der Versuchsreihen präsentiert und diskutiert werden.

¹³FoundationPose wurde in Versuch A1 und A2 zusätzlich ohne das Objekt-Tracking getestet.

5. Implementierung und praktische Durchführung

Im vorliegenden Kapitel erfolgt eine detaillierte Beschreibung der Implementierungsschritte und der praktischen Durchführung einer automatisierten Methode zur annotierten Datenaufnahme für die 6D-Posenbestimmung. Der Schwerpunkt liegt dabei auf der Darstellung der spezifischen technischen und praktischen Herausforderungen, denen wir im Verlauf der Implementierung begegnet sind, sowie auf der Entwicklung von Lösungen zur Überwindung dieser Hürden.

Die Reihenfolge des Kapitels entspricht dabei grob dem in Abbildung 39 dargestellten Ablauf aus Kapitel 3. Zunächst wird auf den praktischen Versuchsaufbau und die Programmierung des Roboters eingegangen. Anschließend wird die Festlegung der relativen Ground-Truth sowie die dafür notwendigen Schritte detailliert beschrieben. Es folgt eine Beschreibung der Implementierung der drei verwendeten Methoden zur 6D-Posenbestimmung. Abschließend wird der gesamte Ablauf dargestellt, der sich aus den vorherigen Schritten für den Testaufbau ergibt.

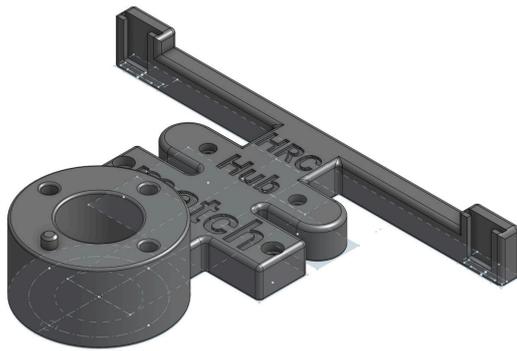
5.1. Praktischer Versuchsaufbau

Der praktische Versuchsaufbau umfasst primär die Kalibrierung und Programmierung des LBR iiwa sowie die Inbetriebnahme der verwendeten Sensoren. Im Folgenden werden einige Aspekte kurz erläutert.

5.1.1. Aufbau und Konfiguration der notwendigen Hardware

Zunächst wurde ein Halter für die Sensoren benötigt, der diese an einer festen Position relativ zum Flansch des Roboters fixiert. Der Halter dient außerdem als Befestigung für zwei zusätzliche Leuchten, um eine bessere Ausleuchtung der Objekte zu gewährleisten. Der finale Zustand des nach mehreren Design-Iterationen 3D-gedruckten Halters ist in Abbildung 51 zu sehen.

Die Positionen der beiden Sensoren auf dem Halter sind so dicht wie möglich beieinander angeordnet, um einen direkten Vergleich der Objekte zwischen beiden Sensoren zu ermöglichen. Zudem wurden alle Kanten mit Radien versehen, um das Verletzungsrisiko zu minimieren, da der Halter an einem Roboter montiert ist, der über keine physische Abgrenzung verfügt. Ein Offset am Roboterflansch wurde hinzugefügt, um die Ergonomie bei der Handführung des Roboters zu verbessern.



(a) CAD-Modell



(b) Montierter Halter mit Sensoren und Lichtern

Abbildung 51: 3D-gedruckter Halter zur Befestigung der Sensoren

Die Kalibrierung eines Base Systems auf dem Roboter wurde entsprechend der Dokumentation von KUKA nach der *3-Punkt-Methode* durchgeführt (vgl. [68]). Das Vorgehen ist in Abbildung 52 schematisch dargestellt.

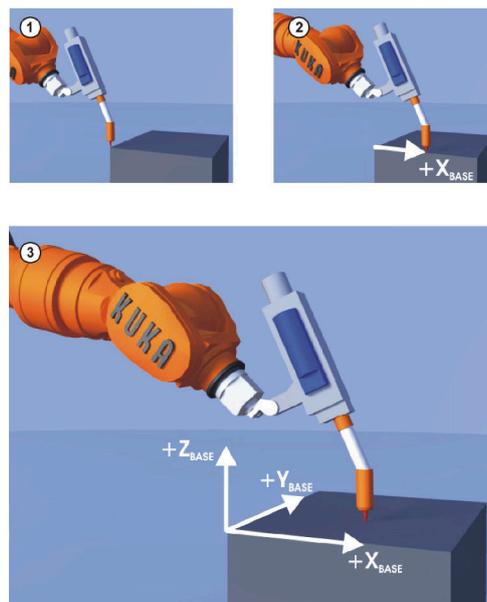
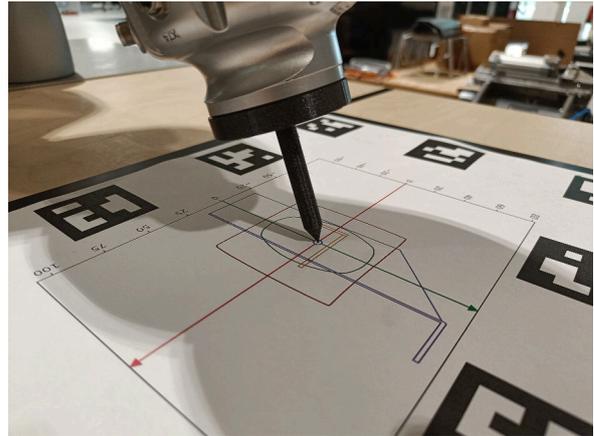


Abbildung 52: 3-Punkt-Methode zum Kalibrieren eines Base-Koordinatensystems [68]

Um diese Methode zur Kalibrierung durchführen zu können, ist zunächst ein Werkzeug mit einem definierten Abstand zum Flansch des Roboters (in der z-Achse des Werkzeugkoordinatensystems) erforderlich. Zu diesem Zweck wurde eine einfache Geometrie mit einer genau festgelegten Länge von 75 mm entworfen und im 3D-Druckverfahren hergestellt. Um mögliche Fertigungstoleranzen des 3D-Druckprozesses zu kompensieren, wurde diese Geometrie zusätzlich mit der *XYZ 4-Punkt Methode* von KUKA kalibriert (vgl. [68]), siehe Abbildung 53a. Anschließend erfolgte die Kalibrierung des Basissystems gemäß den beschriebenen Schritten, siehe Abbildung 53b.



(a) Kalibrierung des Referenzierungswerkzeugs



(b) Kalibrierung der Template Base

Abbildung 53: Kalibrierung des verwendeten KUKA LBR iiwa

5.1.2. Roboterprogramm für die automatisierte Datenaufnahme

Die Aufgabe des Roboterprogramms besteht darin, die drei Bahnen, die in Kapitel 3 beschrieben und in Abbildung 49 dargestellt sind, abzufahren. Die Geschwindigkeit soll dabei konstant, aber anpassbar sein. Zudem soll während der jeweiligen Bewegungen die aktuelle Ist-Pose des Roboters übertragen werden, um diese zur Bestimmung der Ground-Truth verwenden zu können.

Bei der Programmierung der Roboterbewegungen stellten der begrenzte Arbeitsraum und die limitierte Manipulierbarkeit des Roboters eine Herausforderung dar. Da die Bewegungen einen möglichst großen Sichtbereich auf das Objekt abdecken sollen, dessen Pose bestimmt werden soll, befanden sich einzelne Punkte der Bewegungen am Rande des Arbeitsraumes. Weiterhin ist die Manipulierbarkeit des LBR iiwa verhältnismäßig stark eingeschränkt, da zwischen den einzelnen Achsen aus Sicherheitsgründen stets genug Platz bleiben muss, um etwaige Einklemmungen zu vermeiden. Insbesondere stellten jedoch die linearen- und Kreisbewegungen eine Herausforderung dar, da während dieser Bewegungen eine kontinuierliche Orientierung des Roboters zur Datenaufnahme erforderlich ist. In Summe war die Programmierung der Bewegungen deshalb mit einem großen Maß an Versuch und Irrtum verbunden¹⁴.

Für die Programmierung wurde die Software *KUKA Sunrise.Workbench 1.15.0.9* verwendet, welche die Programmierung des Roboters in Java ermöglicht. Als Betriebssystem auf dem LBR iiwa läuft das korrespondierende *KUKA Sunrise.OS 1.15.0.9*.

¹⁴Es wäre erstrebenswert, eine Methode zu entwickeln, die die optimale Positionierung des Roboters unter Vorgabe von Bahn und Orientierung automatisch bestimmt.

Die kontinuierliche Übertragung der Ist-Pose des Roboters erfolgt mithilfe der asynchronen Funktion *moveAsync*, siehe Quellcode 3.

```

165  robot.move(ptp(Right).setJointVelocityRel(ptpVel));
166  // Small delay before start of motion
167  try {
168      Thread.sleep(sleepTime);
169  } catch (InterruptedException e) {
170  }
171
172  motion = robot
173      .moveAsync(circ(Center, Left).setCartVelocity(CaptureVel));
174
175  transmitpose(motion, sampleTime, "circ_far");

```

Quellcode 3: Bewegungsbefehle des Roboters zur Datenaufnahme der Objekte

In Zeile 165 wird die konventionelle PTP-Bewegung zur Startposition der Kreisbahn durchgeführt. Nach einer kurzen Verzögerung, um das Ausschwingen abzuschließen, wird die asynchrone Bewegung gestartet, die einen *IMotionContainer* erzeugt. Dieser Container enthält Informationen über die Bewegung und wird anschließend in der Funktion *transmitpose* verwendet, um das Ende der Bewegung abzufragen.

Für die Übertragung der Ist-Pose in der Funktion *transmitpose* bestand zunächst das Problem, dass die übermittelte Pose nicht ausreichend genau war. Die Translation wurde auf zwei Nachkommastellen in Millimetern und die Orientierung auf zwei Nachkommastellen in Radianten angegeben. Diese geringe Auflösung resultierte in einer „hakenden“ Trajektorie, welche für die Festlegung der Ground-Truth verwendet wurde. Das Problem konnte dadurch behoben werden, dass der Frame der *currentPose* nicht direkt verwendet wurde, sondern dessen Einzelkomponenten, siehe Quellcode 4.

```

220  Frame currentPose = robot.getCurrentCartesianPosition(
221      robot.getFlange(), Template_Base_ref);
222
223  Double X = currentPose.getX();
224  Double Y = currentPose.getY();
225  Double Z = currentPose.getZ();
226  Double A = currentPose.getAlphaRad();
227  Double B = currentPose.getBetaRad();
228  Double C = currentPose.getGammaRad();

```

Quellcode 4: Bestimmung der genauen Pose des Roboters über die Funktion *getCurrentCartesianPosition*

Die so verwendeten Daten vom Java-Typ *double* haben 15-16 Dezimalstellen und ermöglichen somit eine sehr präzise Beschreibung der Ist-Pose.

5.2. Relative Festlegung der Ground-Truth

In diesem Kapitel sollen ausgewählte Bestandteile des Prozesses zur Festlegung der Ground-Truth relativ zu den am Roboter montierten Sensoren vorgestellt und einige der Herausforderungen beschrieben werden, die dabei auftreten.

5.2.1. Transformation der Koordinatensysteme mit `pytransform3d`

Die Notwendigkeit der Transformation verschiedener Koordinatensysteme ergibt sich in nahezu jedem Schritt dieser Arbeit. Dabei besteht die Herausforderung darin, dass die verwendeten Systeme und Software eine Vielzahl unterschiedlicher Repräsentationen und Konventionen für Koordinaten verwenden. Die Softwarebibliothek `pytransform3d` von Fabisch [69] war daher von unschätzbarem Wert für die erfolgreiche Durchführung dieser Arbeit. Neben einer robusten und umfangreichen Implementierung nahezu aller möglichen Transformationen verfügt diese Bibliothek auch über eine hervorragende Dokumentation¹⁵.

Zur Umrechnung wurde der `transform_manager` genutzt. Dieser ermöglicht die Angabe sämtlicher Transformationen in einem jeweils gemeinsamen Bezugskordinatensystem. Sofern eine Verkettung der Transformationen möglich ist, lassen sich anschließend direkte Transformationen zwischen den einzelnen Systemen ermitteln. Die Umrechnung verschiedener Repräsentationen und Konventionen erfolgt ebenfalls, sodass sämtliche Transformationen einheitlich unter Verwendung der homogenen Transformation durchgeführt werden können. Die Transformation zur Referenzierung der Sensoren, welche im nächsten Abschnitt erläutert wird, ist beispielhaft in Abbildung 54 dargestellt.

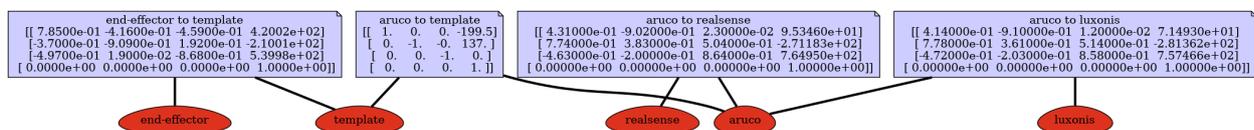


Abbildung 54: Beispiel für die ausgeführten Transformationen

Dargestellt sind die homogenen Transformationen zwischen den einzelnen Komponenten des gesamten Systems. Die Transformation zwischen `end-effector` und `template` wird exemplarisch durch die Ist-Pose des Roboters beschrieben. Mit dem `transform_manager` lassen sich anschließend auch nicht explizit gegebene Transformationen bestimmen, etwa von `realsense` zu `end-effector`.

¹⁵<https://dfki-ric.github.io/pytransform3d/>

Neben der bereits beschriebenen Transformationen erwies sich pytransform3d auch als außerordentlich nützlich für das Debugging der entwickelten Anwendung. Diesbezüglich sei darauf verwiesen, dass eine Visualisierung der Transformationen in einem Bezugssystem möglich ist, wie in Abbildung 55 dargestellt.

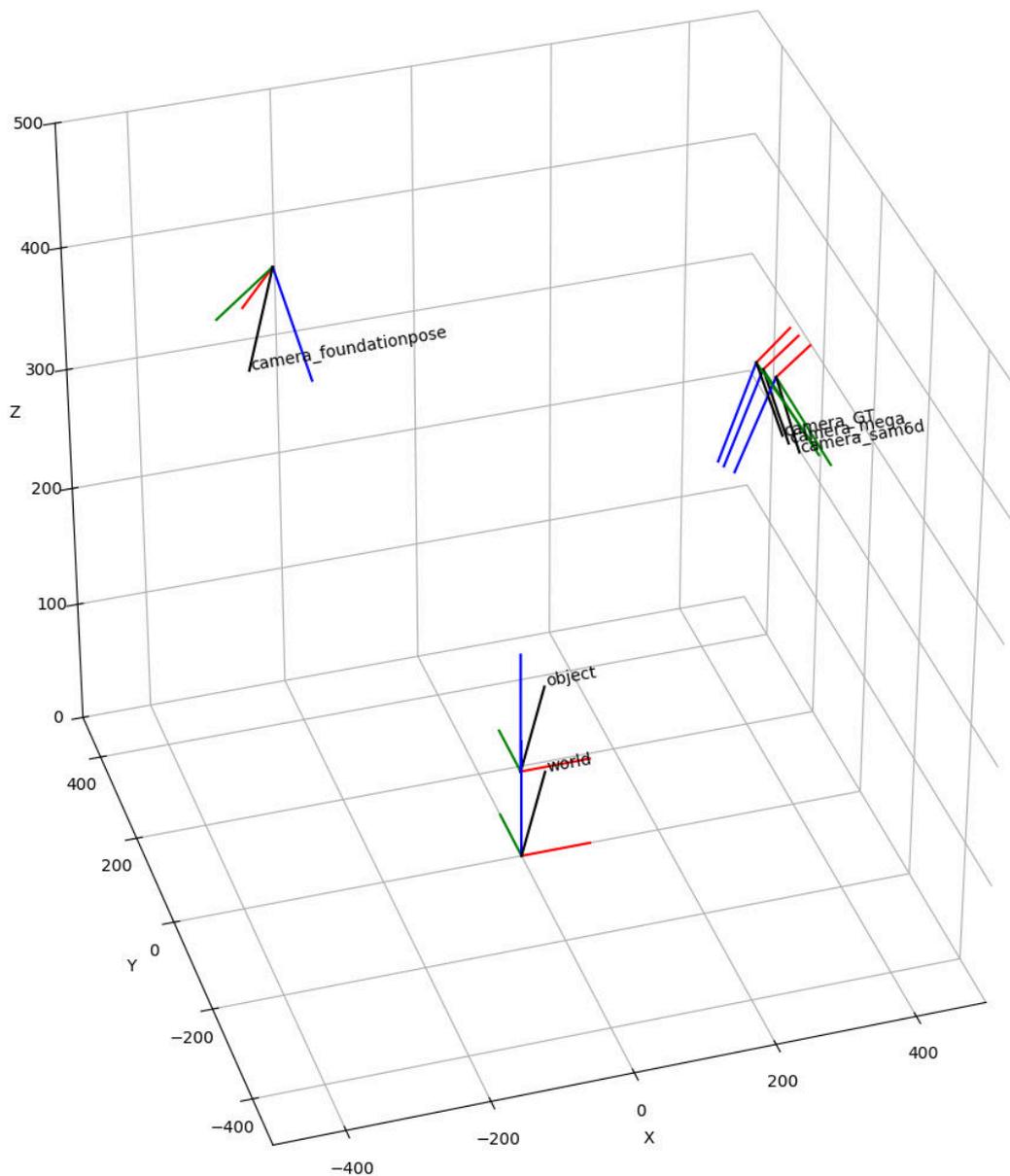


Abbildung 55: Beispiel für das plotten mehrerer Posen mit pytransform3d

Zu sehen ist die Posenbestimmung eines Objektes (*object*) im Template-Koordinatensystem (*world*). Die Koordinatensysteme in positiver z-Richtung beschreiben die Pose der Ground-Truth sowie die Ergebnisse der drei Methoden zur Posenbestimmung. Wie zu erkennen ist, liegt in diesem Beispiel die Pose der FoundationPose-Methode (*camera_foundationpose*) weit abseits der anderen, was auf einen Fehler bei einer der Transformationen hindeutet.

5.2.2. Referenzierung der Sensoren über ArUco Marker

Basierend auf den Erkenntnissen der vorherigen Abschnitte erfolgt die Referenzierung der Sensoren relativ zu dem Flansch des Roboters. Ziel ist es, die (starre) Transformation zwischen Flansch und Sensor zu ermitteln, da diese in Kombination mit der Ist-Pose des Roboters während der Datenaufnahme zukünftig als Teil der Ground-Truth verwendet wird.

Im Rahmen dieser Referenzierung wurden ArUco-Marker verwendet. Hierbei handelt es sich um visuelle Schwarz-Weiß-Marker mit einem einzigartigen binären Muster, welches eine eindeutige Identifikation von Objekten sowie die Bestimmung ihrer Position und Orientierung im Raum ermöglicht. Ihre Anwendung findet sich in den Bereichen Augmented Reality, Roboternavigation und Kamerakalibrierung.

Ist die Lage der ArUco-Marker im Basiskoordinatensystem des Roboters fest und definiert, lässt sich diese über eine Transformation $ArUco_{Roboter}$ ausdrücken. Bestimmt werden soll die Transformation des Sensors relativ zum Roboter $Sensor_{Roboter}$. Dies geschieht, indem zunächst die Lage des ArUco-Markers im Koordinatensystem der Kamera $ArUco_{Sensor}$ ermittelt wird. Anschließend kann $Sensor_{Roboter}$ über die Differenz der beiden bekannten Transformationen bestimmt werden, siehe Abbildung 56.

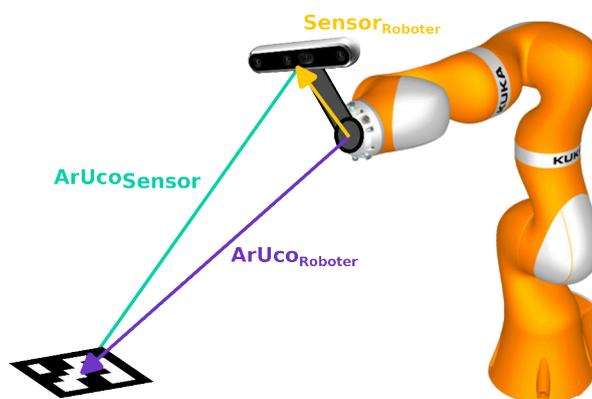


Abbildung 56: Veranschaulichung der Referenzierung eines Sensors über ArUco-Marker

Die Implementierung der Posenbestimmung der ArUco-Marker basiert dabei auf der offiziellen *OpenCV 4.10* Dokumentation [70]. Quellcode 5 zeigt die Festlegung des Gitters aus ArUco-Markern, das im Rahmen des Templates (vgl. Abschnitt 5.2.3) genutzt wird. Hierzu wird die Funktion *GridBoard* verwendet, welche neben der Anzahl der Marker im Gitter, deren Größe und dem Abstand zueinander auch die verwendete Sammlung von ArUco-Markern benötigt (vgl. [71]).

```

154 aruco_dict = cv.aruco.getPredefinedDictionary(cv.aruco.DICT_5X5_50)
155 board = cv.aruco.GridBoard((4, 3), 50, 65, aruco_dict)
156 # board_image = board.generateImage((1000, 1000), None, 0, 1)
157
158 detectorParameters = cv.aruco.DetectorParameters()
159 detector = cv.aruco.ArucoDetector(aruco_dict, detectorParameters)

```

Quellcode 5: Festlegung des verwendeten ArUco-Gitters mit OpenCV

Das verwendete Board besteht aus 4x3 Markern mit einer Größe von 50 mm und einem Abstand von 65 mm. Als Sammlung wird *ArUco 5x5_50* verwendet. Diese Sammlung bildet mit 5x5 Bits 50 verschiedene Codes ab, die einen minimalen Hamming-Abstand von 8 haben. Sie wurde verwendet, da sie einen guten Kompromiss aus fehlerfreier Lesbarkeit aufgrund der Hamming-Distanz und guter Erkennbarkeit aufgrund der relativ geringen Bit-Anzahl bildet.

Unter Verwendung dieser Sammlung kann anschließend der *ArucoDetector* initiiert werden. Dieser wird anschließend für die Erkennung der Marker verwendet, siehe Quellcode 6.

```

211 # Detect markers
212 corners, ids, rejectedImgPoints = detector.detectMarkers(image)
213 # if at least one marker detected
214 if not (ids is None) and len(ids) > 0:
215     cv.aruco.drawDetectedMarkers(image_copy, corners, ids)
216
217 # Get object and image points for the solvePnP function
218 objPoints, imgPoints = board.matchImagePoints(corners, ids)
219
220 # Find pose
221 retval, rvec, tvec, inliers = cv.solvePnPRansac(
222     objPoints, imgPoints, cam_matrix, dist_coefficients
223 )

```

Quellcode 6: Erkennung und Posenbestimmung des ArUco-Gitters mit OpenCV

Der zuvor initialisierte *detector* erhält das Bild als Eingabe und gibt eine Liste aller erkannten Marker mit deren Nummern (*ids* 0-11) und den jeweiligen Eckpunkten im Bild (*corners*) zurück. Unter Verwendung der Eckpunkte und des zuvor definierten ArUco-Gitters (*board*) lässt sich nun die Korrespondenz zwischen den Koordinaten herstellen. Hierfür wird die *solvePnPRANSAC*-Funktion genutzt, welche das Perspective-n-Point (PnP)-Verfahren mit Random Sample Consensus verwendet, um die Translation (*tvec*) und Rotation (*rvec*) des ArUco-Gitters im Koordinatensystem der Kamera zu berechnen. Wie in Abschnitt 4.1 beschrieben, ist es für diese Berechnung weiterhin notwendig, die intrinsischen Kameraparameter (*cam_matrix*) sowie etwaige Verzeichungskoeffizienten (*dist_coefficients*) anzugeben.

Die berechneten Daten können anschließend unter Verwendung weiterer OpenCV-Funktionen visualisiert werden, siehe Abbildung 57.

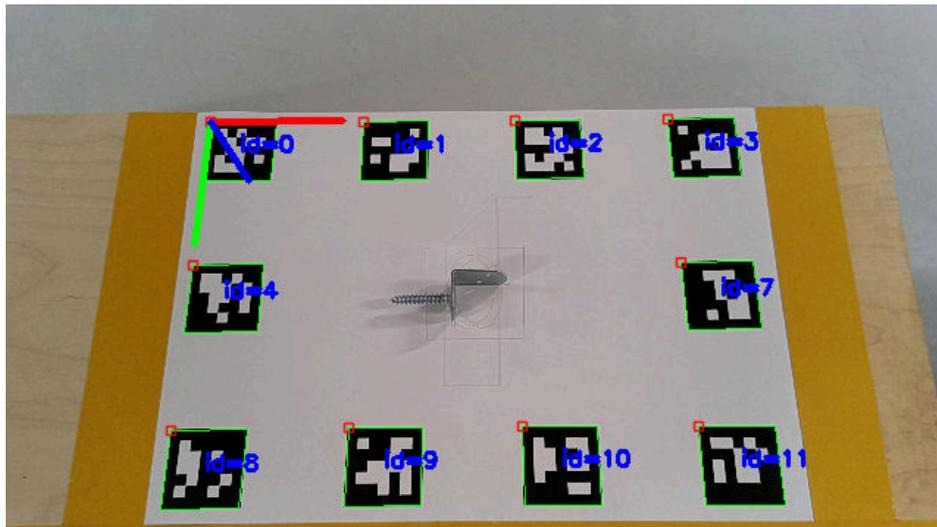
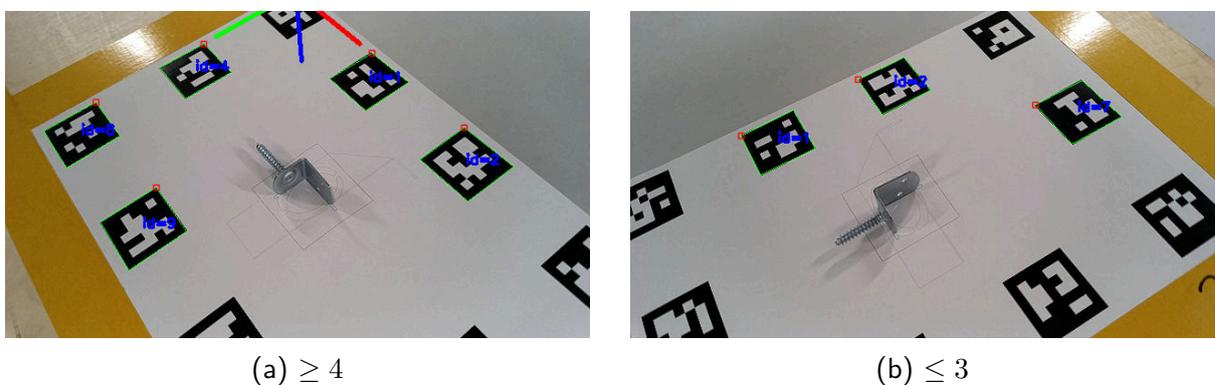


Abbildung 57: Beispiel für die ArUco Posenbestimmung: Detektierte Kanten der Marker, deren IDs, sowie die Pose des ArUco-Gitters sind visualisiert

Zu erkennen sind die 12 detektierten ArUco-Marker mit deren IDs in Blau. Der grüne Rahmen bildet die erkannten Eckpunkte der einzelnen Marker ab. Für die Posenbestimmung des Grids werden die oberen linken Ecken jedes Markers verwendet, dargestellt in Rot. Die bestimmte Pose bezieht sich, wie dargestellt, auf den Marker mit der ID 0, wobei die z-Achse in das ArUco-Gitter zeigt.

Die Verwendung eines ArUco-Gitters anstelle eines einzelnen Markers für die Posenbestimmung ist wesentlich robuster. Einerseits durch die vergleichsweise großen Abstände der Punkte zueinander, welche für die Berechnung genutzt werden. Andererseits ist das Verfahren auch robust gegenüber der Verdeckung einzelner Marker. Sofern die Anzahl erkannter ArUco-Marker größer oder gleich 4 ist, lässt sich die Posenbestimmung ausführen. Dies gilt auch, wenn die Pose des ArUco-Gitters außerhalb des sichtbaren Bildbereichs liegt, vgl. Abbildung 58.



(a) ≥ 4

(b) ≤ 3

Abbildung 58: Beispiele für notwendige Mindestanzahl an sichtbaren ArUco-Markern für die Posenbestimmung eines ArUco-Gitters mit OpenCV

Um die Genauigkeit der Transformation $ArUco_{Sensor}$ zu erhöhen und etwaige Abweichungen bei der Posenbestimmung der ArUco-Marker zu kompensieren, soll diese mehrmals durchgeführt werden, und anschließend wird ein Mittelwert gebildet. Da die Berechnung eines Mittelwertes für eine Rotationsmatrix jedoch nicht direkt möglich ist, wurde diese zunächst in Quaternionen umgerechnet. Anschließend wurde das Verfahren von Markley u. a. [72] zur Berechnung der Quaternion-Mittelwerte verwendet¹⁶.

5.2.3. Verwendung von Templates für die Ground-Truth

Wie zuvor mehrfach erwähnt, sollen Templates verwendet werden, um die Position und Orientierung der Objekte relativ zum Roboter sicherzustellen. Zu diesem Zweck sollen von den 3D-Modellen der Objekte Fußabdrücke als Objekt-Referenzmarkierungen erstellt werden. Die Footprints aller Objekte sollen anschließend auf Templates gedruckt werden, welche für die automatisierte Datenaufnahme verwendet werden. Somit ist die Lage des Objektkoordinatensystems relativ zum Koordinatensystem des Templates, welches wiederum als Basissystem des Roboters verwendet wird, bestimmt (vgl. Abschnitt 5.2.1).

Um die Objekt-Referenzmarkierungen zu erstellen, wurde die Software *trimesh* [73] verwendet. Abbildung 59 zeigt die unbearbeiteten Objekt-Referenzmarkierungen der vier verwendeten Objekte.

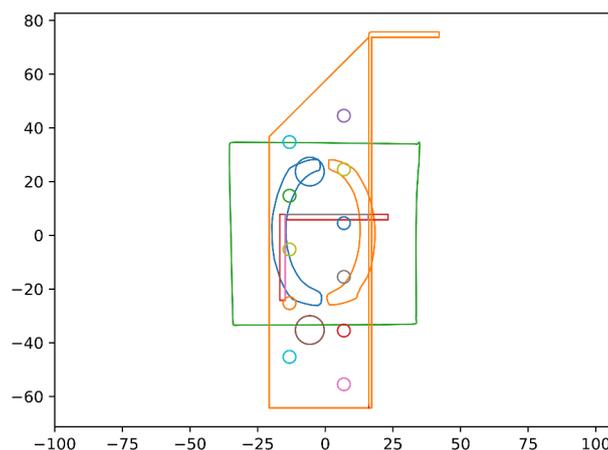


Abbildung 59: Ausgabe des Programms zum Erstellen der Objekt-Referenzmarkierungen

Das Programm zum Erstellen der Objekt-Referenzmarkierungen unter Verwendung von *trimesh* ist exemplarisch in Quellcode 7 aufgeführt.

¹⁶Python-Implementierung basiert auf <https://github.com/christophhagen/averaging-quaternions>.

```

19 mesh1 = trimesh.load_mesh("object_models/obj_000002.ply")
20 slice1 = mesh1.section(plane_origin=[0, 0, -73.5], plane_normal=[0, 0, 1])
21 slice_2D1, to_3D1 = slice1.to_planar()

```

Quellcode 7: Erstellen der Schnittebene der 3D-Modelle mit trimesh

Zunächst wird das 3D-Modell des Objektes geladen, anschließend wird die Schnittebene festgelegt. Diese entspricht weitestgehend den Grenzen des 3D-Modells in negativer z-Richtung. Es kann jedoch notwendig sein, die Orientierung der Schnittebene zunächst anzupassen, damit es einen festen Stand für die Datenaufnahme hat. Anschließend wird die *section* Funktion verwendet. Dieser Schnitt kann mit der Funktion *to_planar* in eine Liste mit allen Punkten des Modells in der Schnittebene (*slice_2D1*), sowie der homogenen Transformation vom Objektkoordinatensystem zur Schnittebene (*to_3D1*) umgewandelt werden.

Für eine bessere Erkennbarkeit der Objekt-Referenzmarkierungen werden diese abschließend manuell angepasst, um unnötige Objektdetails auszublenden. Das finale Template für die Datenaufnahme, welches das ArUco-Gitter zur Referenzierung beinhaltet, ist in Abbildung 60 dargestellt.

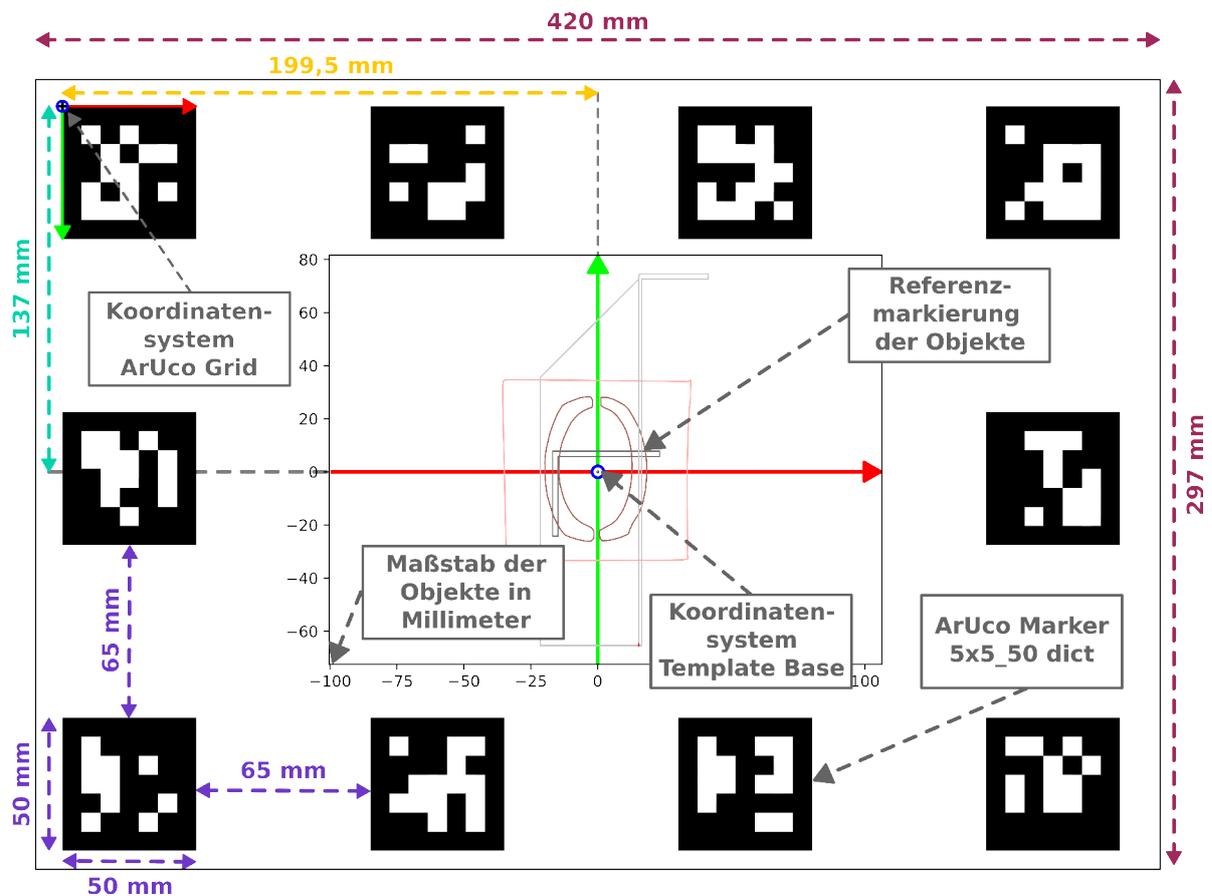


Abbildung 60: Darstellung eines Templates mit ArUco-Markern, Koordinatensystemen, Referenzmarkierungen der Objekte und Beschriftung, sowie Bemaßung

5.3. Implementierung der Methoden zur 6D-Posenbestimmung

Als letzter Schritt der Implementierung soll in diesem Kapitel auf einige Herausforderungen und besonders relevante Aspekte eingegangen werden, die sich bei der Implementierung der drei verwendeten Methoden zur 6D-Posenbestimmung ergeben haben.

5.3.1. Automatische Generierung der Objektmasken mit BlenderProc

Die 6D-Posenbestimmung neuartiger starrer Objekte erfordert, wie in Abschnitt 2.4 beschrieben, in der Regel eine Region of Interest, welche grob spezifiziert, wo im Bild sich das zu bestimmende Objekt befindet. Eine Ausnahme bildet die SAM6D-Methode, die das SAM-Modell nutzt, um die Region of Interest zu generieren. Für die praktische Implementierung bedeutet dies, dass eine Methode entwickelt werden muss, die zusätzlich zu den aufgenommenen Daten auch eine Region of Interest – oder im Falle von FoundationPose eine Objektmaske – als zusätzliche Eingabe für die Posenbestimmung liefert, vgl. Abbildung 61.

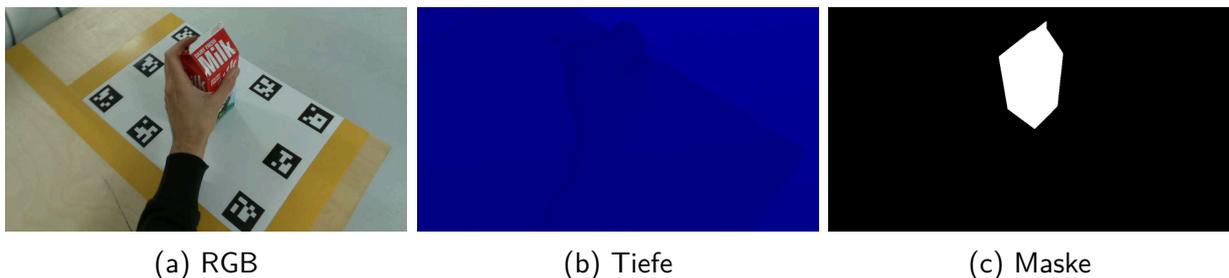


Abbildung 61: Aufgenommene Daten und mit BlenderProc erstellte Objektmaske

Zu diesem Zweck wurde die Software *BlenderProc* vom Institut für Robotik und Mechatronik des Deutschen Zentrums für Luft- und Raumfahrt (DLR) verwendet [74, 75]. Bei *BlenderProc* handelt es sich um eine Pipeline für die 3D-Rendering-Software *Blender*, deren Hauptanwendung in der Erstellung synthetischer Datensätze liegt. Aufgrund der weiten Verbreitung im Forschungsbereich der 6D-Posenbestimmung umfasst *BlenderProc* jedoch auch Funktionen zum Annotieren von manuell aufgezeichneten Daten mit der Ground-Truth.

Unter Verwendung von *BlenderProc* wird über alle aufgenommenen Bilder iteriert, und anhand der mit *pytransform3d* berechneten Transformationen zunächst die Lage der Objekte im Bild auf Basis der Ground-Truth berechnet. Anschließend können unter Verwendung der intrinsischen Kameraparameter die Objektmasken gerendert werden. Dieses Vorgehen muss für jeden Sensor separat erfolgen.

Weitere Ausgaben von BlenderProc umfassen die Ground-Truth-Daten im jeweiligen Koordinatensystem der Kamera, siehe Quellcode 8.

```
12 "10": [{"cam_R_m2c": [0.7513357400894165, 0.6592817902565002, -0.02901923656463623,
↪ 0.5595933794975281, -0.6598027944564819, -0.5015131235122681, -0.3497854769229889,
↪ 0.36056575179100037, -0.8646631240844727], "cam_t_m2c": [0.1261751651763916, -76.86406707763672,
↪ 478.4615478515625], "obj_id": 1}],
```

Quellcode 8: Beispiel für BlenderProc Ground-Truth Pose aus scene_gt.json

Die Daten umfassen die Rotationsmatrix, die Translation als Vektor in Millimetern, sowie die Objekt-ID, welche nur relevant ist, falls sich mehrere Objekte im Bild befinden. Dies ist hier jedoch nicht der Fall.

Eine weitere Ausgabe von BlenderProc ist die Bounding Box, welche die Region of Interest um das Objekt beschreibt, siehe Quellcode 9.

```
12 "10": [{"bbox_obj": [542, 45, 203, 345], "bbox_visib": [542, 45, 203, 345], "px_count_all": 46889,
↪ "px_count_valid": 46889, "px_count_visib": 46864, "visib_fract": 0.9994668259079955}],
```

Quellcode 9: Beispiel für BlenderProc Ground-Truth Bounding Box aus scene_gt_info.json

Diese Daten umfassen die Bounding Box des Objektes sowie Angaben zur Verdeckung des Objektes, wie eine angepasste Bounding Box und Informationen zu sichtbaren und verdeckten Pixeln sowie deren Verhältnis. Die Verdeckung wird dabei nativ nur durch andere Objekte in der Szene berechnet und nicht durch die Hand, welche sich in diesem Fall teilweise vor dem Objekt befindet¹⁷ (vgl. Abbildung 61).

5.3.2. Angleichung der RGB-D Daten

Im Falle der Aufnahme von RGB-D-Daten als Eingabe für die 6D-Posenbestimmung ist darauf zu achten, dass die RGB- und Tiefen-Frames aneinander angeglichen sind. Ist dies nicht der Fall, kann es zu Komplikationen oder signifikanten Ungenauigkeiten bei der Posenbestimmung kommen. Durch die unterschiedliche Position der jeweiligen Sensoren innerhalb der Kamera sowie jeweils unterschiedliche intrinsische Kameraparameter kommt es dazu, dass beide Sensoren unterschiedliche Perspektiven und Bildausschnitte beinhalten. Ein Beispiel für einen RGB- und Tiefen-Frame, welche zum selben Zeitpunkt vom selben Sensor aufgenommen wurden, ist in Abbildung 62 zu sehen.

¹⁷Die Erkennung der Hand und die Ableitung eines exakten Verdeckungsverhältnisses werden im Ausblick behandelt, siehe Kapitel 7.

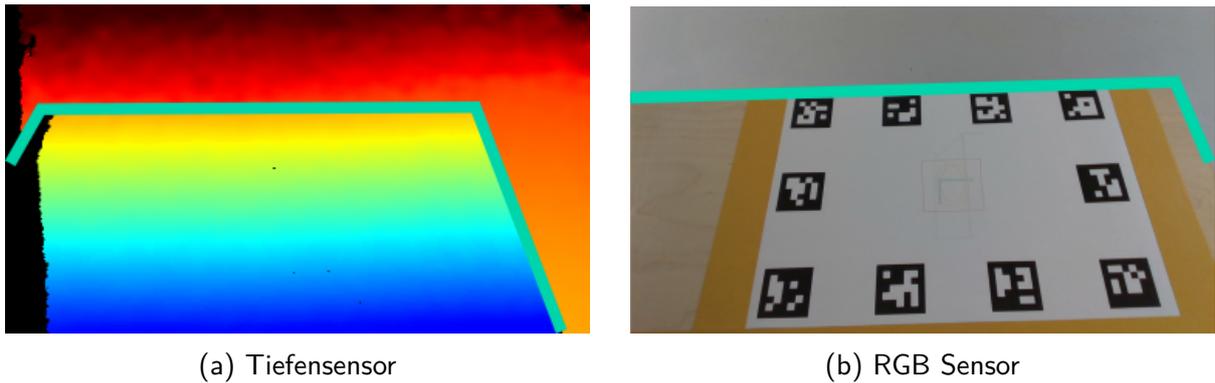


Abbildung 62: Beispiel für RGB und Tiefenbild der RealSense D435i ohne Angleichung: Die Kante des Tisches ist zur besseren Unterscheidung farblich hervorgehoben

Wie an der farblich hervorgehobenen Tischkante zu erkennen ist, unterscheiden sich die beiden Bildausschnitte signifikant voneinander. Je nach Art und Abstand der verwendeten Sensoren kann die Angleichung der Frames eine komplexe Aufgabe darstellen. Die verwendeten Sensoren bieten jedoch den Vorteil, dass schon vorhandene Beispiele für die Angleichung sowohl für den Sensor von RealSense¹⁸ als auch für den Sensor von Luxonis¹⁹ verfügbar sind. Diese Beispiele konnten nach geringfügiger Anpassung für die Datenaufnahme verwendet werden.

Eine weitere mögliche Herausforderung kann die zeitliche Anpassung der Frames darstellen. Im vorliegenden Versuchsaufbau wird die Ist-Pose des Roboters zur Festlegung der Ground-Truth verwendet. Es ist somit sicherzustellen, dass die Ist-Pose und die aufgenommenen Frames zeitlich aufeinander abgestimmt sind. Die generelle Problematik ist in Abbildung 63 schematisch dargestellt.

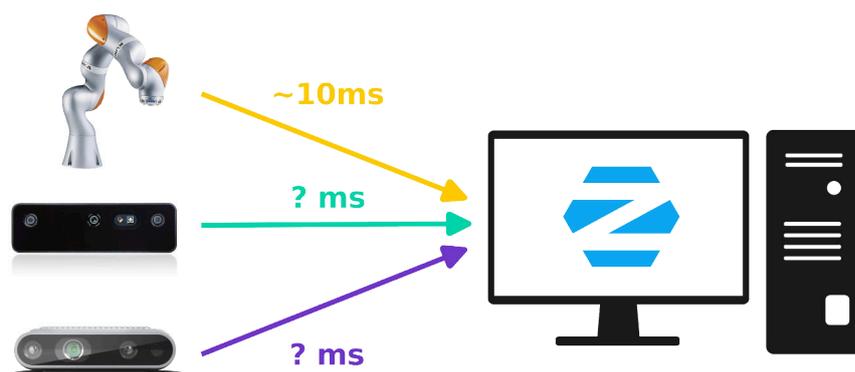


Abbildung 63: Schematische Darstellung der Zeitverzögerung der Sensorsignale

¹⁸<https://github.com/IntelRealSense/librealsense/blob/master/wrappers/python/examples/align-depth2color.py>

¹⁹<https://github.com/luxonis/depthai-python/blob/main/examples/StereoDepth/rgbdepthaligned.py>

Sollte die zeitliche Differenz zwischen der Ground-Truth und den Sensor-Frames zu groß sein, stimmt die Position des Objekts im Bild nicht mehr mit der Ground-Truth überein. Das Signal vom LBR iiwa zum verwendeten Computer zur zentralen Steuerung und Datenverarbeitung weist eine konstante Latenz von etwa 10 ms auf. Diese Latenz wurde gemessen, indem der Roboter auf ein Startsignal des Computers wartet und nach 10 ms die erste Ist-Pose sendet. Über einen Parameter im Programm werden diese Posen fortschreitend in Intervallen von 10 ms gesendet. Beim Empfang der Sensor-Signale kann es jedoch zu größeren Verzögerungen kommen. Zu Beginn der Versuche wurde ein leistungsärmerer Computer verwendet, was dazu führte, dass das Empfangen und Speichern der Frames bis zu 3000 ms dauerte. Ein Hilfsverfahren zum Messen der Sensorlatenz wurde entwickelt. Nachdem jedoch ein leistungsstärkerer Rechner verwendet wurde, betrug die Latenz relativ konstant 100 ms. Die mit dem Messverfahren ermittelte Latenz wird von den Zeitstempeln der Frames subtrahiert, und anschließend wird die zeitlich nächstgelegene Ist-Pose zur Festlegung der Ground-Truth bestimmt.

5.3.3. MegaPose

Die MegaPose-Methode wurde unter Verwendung des bereitgestellten *run_inference*-Skripts in einer leicht angepassten Version eingesetzt. Hierbei ist zu beachten, dass die Methode 3D-Modelle im Millimeter-Maßstab erwartet. Die intrinsischen Kameraparameter müssen in der in Quellcode 10 aufgeführten Form bereitgestellt werden.

```
1  {"K": [[514.879638671875, 0.0, 318.87933349609375], [0.0, 514.879638671875, 182.59268188476562],  
   ↪  [0.0, 0.0, 1.0]], "resolution": [360, 640]}
```

Quellcode 10: Inhalt der camera_data.json für MegaPose

Hierbei handelt es sich um die Kameramatrix sowie die Auflösung der verwendeten Bilder.

Bei der Verwendung der Modelle des ITODD-Datensatzes trat das Problem auf, dass diese nicht ohne Anpassung genutzt werden konnten. MegaPose versucht nativ, die geladenen Modelle auf eine gewisse Anzahl von Punkten zu reduzieren, um den notwendigen Rechenaufwand konstant zu halten und die Verwendung komplexer 3D-Modelle zu ermöglichen. Im Fall der ITODD-Modelle liegt die Anzahl der Punkte jedoch unter diesem Schwellenwert, wodurch die Reduktion nicht ausgeführt werden kann. Als Lösung wurden die ITODD-Modelle mittels einer Subdivision Surface modifiziert. Diese Technik teilt die Flächen eines Netzes in kleinere Flächen auf, um ein glatteres Aussehen zu erzielen. Dadurch konnte die Methode auch für die Modelle des HOPE-Datensatzes weiterhin ohne Abweichung vom ursprünglichen Verhalten verwendet werden.

Abbildung 64 zeigt die standardmäßige Visualisierung der Ergebnisse mit MegaPose.



Abbildung 64: Beispiele für Visualisierung der Posenbestimmung mit MegaPose

Die ROI visualisiert die Eingabe, die für die Posenbestimmung des Objekts verwendet wurde. Die Kontur stellt den Umriss des 3D-Modells in der erkannten 6D-Pose dar.

MegaPose gibt die geschätzte 6D-Pose im in Quellcode 11 exemplarisch aufgeführten Format aus.

```
1 {"000010": [{"label": "obj_000021", "TWO": [[ 0.7043869905585143, 0.41955842508004315,
  ↪ 0.571511739954032, -0.033645408313390954], [-0.01913599669933319, -0.05807572975754738,
  ↪ 0.506365180015564]]]}
```

Quellcode 11: Ausgabe der geschätzten Pose von MegaPose

Hierbei handelt es sich um die Rotation in Quaternionen und die Translation als Vektor in Metern. Es ist wichtig zu beachten, dass die Quaternionen die $xzyw$ Schuster/JPL-Konvention verwenden, anstatt der ansonsten nahezu ausschließlich genutzten $wxyz$ Hamilton-Konvention [76]. Dies gilt es bei der anschließenden Transformation zur Evaluierung zu berücksichtigen.

5.3.4. SAM6D

Die Implementierung der SAM6D-Methode gestaltet sich aufgrund der Notwendigkeit, die PointNet-Software-Bibliothek zunächst kompilieren zu müssen, auf vielen Systemen als etwas schwieriger. Nach der erfolgreichen Implementierung erwartet SAM6D die Modelle im Millimeter-Maßstab sowie die intrinsischen Parameter in der in Quellcode 12 aufgeführten Form.

Bei der `depth_scale` handelt es sich um den Tiefenmaßstab, welcher angibt, wie vielen Millimetern die Werte im Tiefenframe entsprechen.

```

1  {"cam_K": [514.879638671875, 0.0, 318.87933349609375, 0.0, 514.879638671875, 182.59268188476562, 0.0,
↵  0.0, 1.0], "depth_scale": 1.0}

```

Quellcode 12: Inhalt der camera.json für SAM6D

Die Visualisierung der Ergebnisse beider Teilschritte der SAM6D-Methode ist in Abbildung 65 dargestellt.

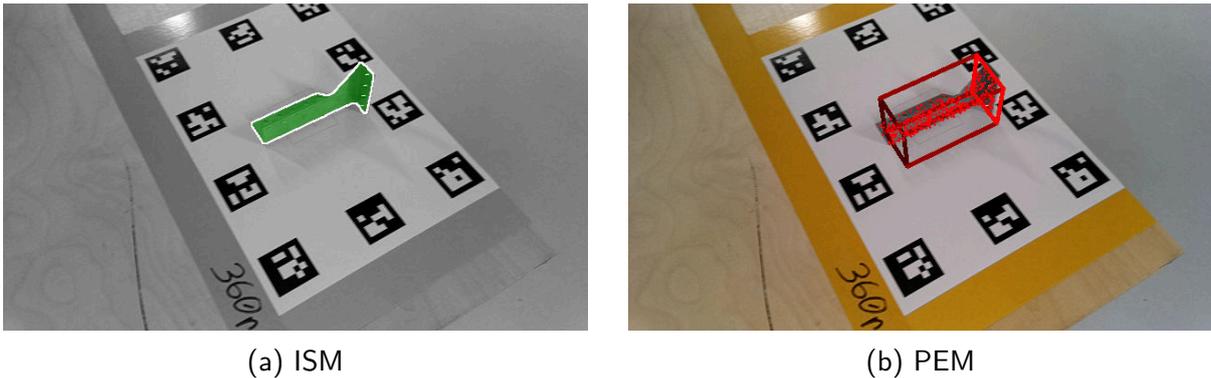


Abbildung 65: Beispiel für Visualisierung der Posenbestimmung mit SAM6D

ISM stellt die Segmentierungsmaske dar, anhand derer die Objekterkennung erfolgt (vgl. SAM-Modell in Abschnitt 2.5.2). Diese Maske wurde nicht unter Verwendung des 3D-Modells erstellt. *PEM* zeigt das Ergebnis des Pose Estimation Model mit der geschätzten 6D-Pose des Objekts. Zu erkennen sind weiterhin die Punkte, die PointNet zur Schätzung der verfeinerten Objekt Pose verwendet.

SAM6D gibt als Ausgabe sämtliche erkannte Segmentierungsmasken zurück. Für jede Maske wird die Posenschätzung mit dem PEM durchgeführt und schließlich die Pose mit dem höchsten Wert als 3x3-Rotationsmatrix und 3x1-Translationsvektor in Millimetern zurückgegeben.

Bei der Verwendung von SAM6D trat mehrfach das Problem auf, dass während der Posenbestimmung Out of Memory Fehler auftraten, da mehr Grafikspeicher (VRAM) angefordert wurde, als im verwendeten System vorhanden war. Um diesen Fehler zu beheben, wurde die Anzahl der Punkte, die das PEM zur Schätzung der Pose verwendet, reduziert²⁰, siehe Quellcode 13.

```

1  points_per_side: Optional[int] = 32,
2  points_per_batch: int = 64,

```

Quellcode 13: Anpassung der im PEM verwendeten Anzahl von Punkten

²⁰Siehe auch <https://github.com/JiehongLin/SAM-6D/issues/24>

Anstelle der Werte 32 und 64 wurden für die durchgeführten Versuche die Werte 16 und 32 verwendet. Dies ermöglicht die Posenbestimmung mit SAM6D bei einer Auflösung von bis zu 720p und 24 GB VRAM möglich.

5.3.5. FoundationPose

Bei der Implementierung von FoundationPose besteht die Besonderheit, dass diese nicht auf einer virtuellen Python-Umgebung basiert, wie MegaPose und SAM6D, sondern unter Verwendung eines virtualisierten Containers mit der Software Docker erfolgt.

FoundationPose erwartet die 3D-Modelle im Meter-Maßstab sowie die Kameraparameter in dem in Quellcode 14 aufgeführten Format.

```
1 {514.879638671875 0.0 318.87933349609375
2 0.0 514.879638671875 182.59268188476562
3 0.0 0.0 1.0}
```

Quellcode 14: Inhalt der cam_K.txt für FoundationPose

Die Visualisierung der FoundationPose-Methode ist beispielhaft in Abbildung 66 dargestellt.

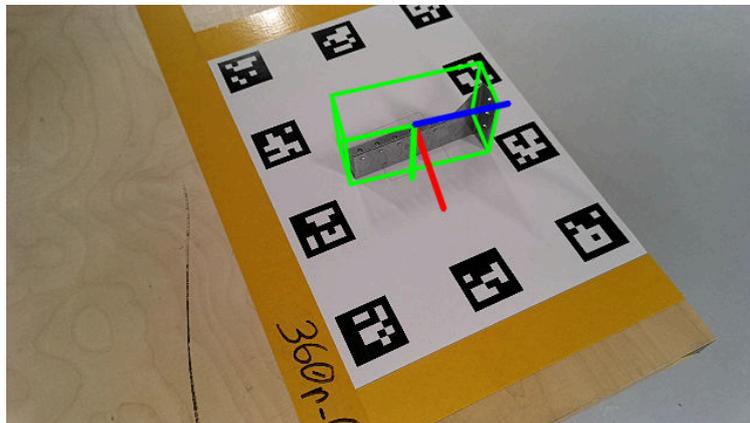


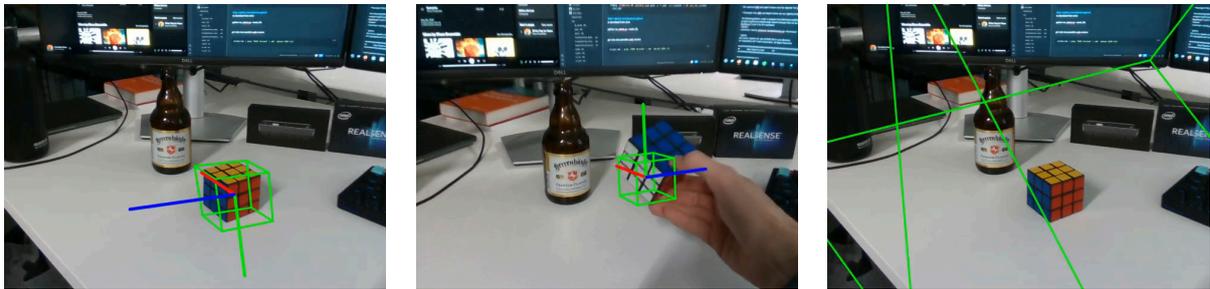
Abbildung 66: Beispiel für Visualisierung der Posenbestimmung mit FoundationPose

Die Ausgabe der Posenbestimmung von FoundationPose erfolgt als homogene Transformation in einer 4x4-Matrix, wobei die Translation in Metern angegeben wird.

Für Vergleichszwecke wurde FoundationPose ohne Objekt-Tracking verwendet. Zu diesem Zweck musste das verwendete *run_demo*-Python-Skript geringfügig angepasst werden²¹.

²¹<https://github.com/NVlabs/FoundationPose/issues/37>

Weitere mögliche Fehler, die bei der Verwendung von FoundationPose sowie bei jeder anderen Methode zur 6D-Posenbestimmung auftreten können, sind in Abbildung 67 dargestellt²².



(a) Falsche intrinsische Parameter (b) Nicht angegliche RGB-D Frames (c) Falscher Maßstab der 3D-Modelle

Abbildung 67: Beispiel für mögliche Fehler bei der 6D-Posenbestimmung

Die Verwendung falscher intrinsischer Kameraparameter kann zu einem geringfügig falsch positionierten oder perspektivisch verzerrten Objekt in der Posenbestimmung führen. Nicht korrekt angegliche RGB-D-Frames können sich auf ähnliche Weise manifestieren. Zusätzlich kann im Falle eines kontinuierlichen Objekt-Trackings das geschätzte Objekt aufgrund der unterschiedlichen Ausrichtung der Frames dem realen Objekt „hinterherhängen“. Weiterhin resultiert ein falscher Objekt-Maßstab, häufig Meter anstatt Millimeter bzw. umgekehrt, in einem signifikant zu großen oder zu kleinen Objekt oder führt sogar dazu, dass keine Posenbestimmung durchgeführt werden kann.

Aufgrund der Erfahrungen, die bei der Implementierung der verschiedenen Methoden gesammelt wurden, lässt sich abschließend konstatieren, dass FoundationPose sowohl die bei weitem leistungsfähigste Methode ist als auch über die beste technische Unterstützung bei Problemen verfügt. Dies liegt einerseits daran, dass sie am häufigsten verwendet wird, aber auch daran, dass der Autor der Methode sich sehr aktiv an Diskussionen beteiligt.

²²<https://github.com/NVlabs/FoundationPose/issues/44>

5.4. Übersicht über den Gesamtprozess

Im Folgenden soll der gesamte Ablauf, dessen Implementierung zuvor beschrieben wurde, abschließend noch einmal beleuchtet werden. Dabei soll insbesondere darauf eingegangen werden, welche Schritte notwendig sind, um die 6D-Posenbestimmung anhand eigener Daten und Metriken zu evaluieren. Dies erfolgt unter Zuhilfenahme des entwickelten Testaufbaus zur automatisierten Datenaufnahme.

Abbildung 68 stellt den gesamten Ablauf grafisch dar. Dabei ist klar ersichtlich, wie die Schritte der *Konfiguration*, *Referenzierung*, *Datenaufnahme*, *Posenbestimmung* und *Evaluierung* aufeinander aufbauen und welche Teilschritte diese jeweils umfassen.

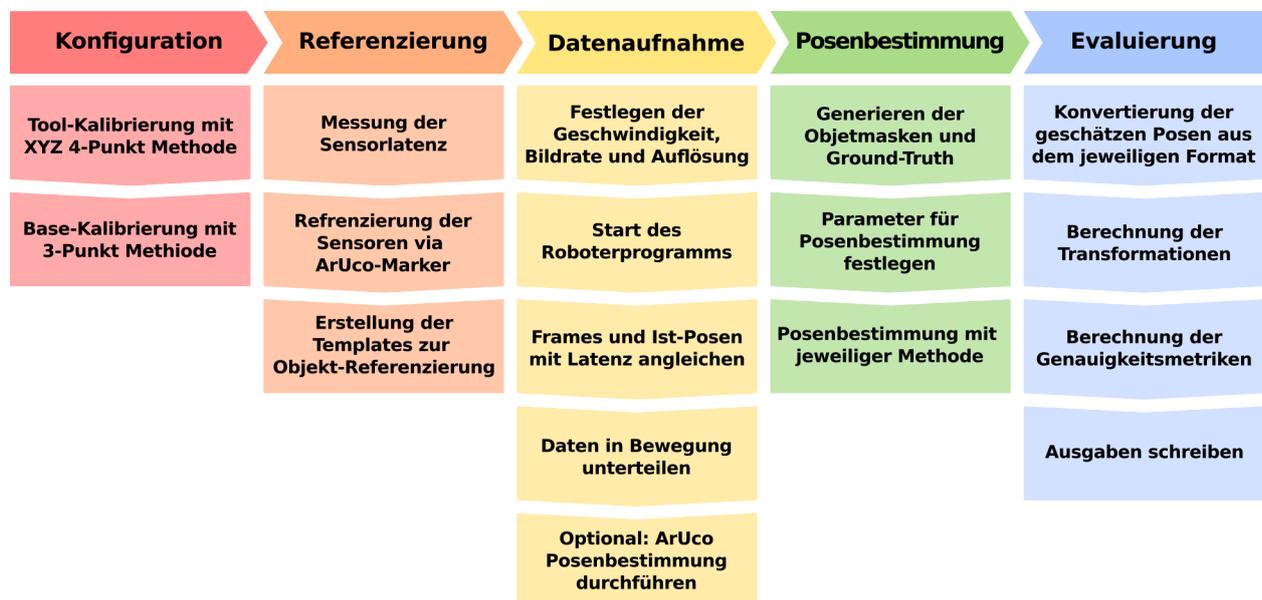


Abbildung 68: Übersicht über den Gesamtprozess mit Teilschritten

Zu Beginn ist es erforderlich, die verwendete Hardware entsprechend zu konfigurieren. Dies umfasst die Kalibrierung eines Referenzierungswerkzeugs, mit welchem anschließend ein Basis-Koordinatensystem für den verwendeten Roboter kalibriert werden kann. Mithilfe dieses Basiskoordinatensystems kann im weiteren Ablauf die Lage der Objekte relativ zum Roboter beschrieben werden.

Es folgt der Prozess der Referenzierung, welcher sicherstellt, dass die Daten, die die Sensoren aufnehmen, akkurat mit der Ground-Truth verknüpft werden können. Hierzu ist sowohl die zeitliche als auch die räumliche Referenzierung der Sensoren erforderlich. Weiterhin werden in diesem Schritt die Templates zur Objekt-Referenzierung erstellt, welche anschließend ausgedruckt zur Referenzierung verwendet werden.

Die Datenaufnahme beginnt mit der Festlegung der jeweiligen Aufnahmeparameter. Anschließend kann mit dem Start des Roboterprogramms die Datenaufnahme beginnen. Die aufgezeichneten Daten werden unter Verwendung der Sensorlatenz an die Ground-Truth, in Form der Roboter-Ist-Posen, angeglichen und in die einzelnen Bewegungen unterteilt. Optional kann eine weitere Posenbestimmung der ArUco-Marker erfolgen, welche die Ground-Truth um eine weitere relative Angabe erweitern kann.

Zum Durchführen der 6D-Posenbestimmung ist es zunächst erforderlich, Objektmasken und die transformierte Ground-Truth mit BlenderProc zu erstellen. Anschließend können die unterschiedlichen Methoden für die Posenbestimmung unter Verwendung zuvor festgelegter Parameter angewendet werden.

Der gesamte Prozess endet mit der Evaluierung der gewonnenen Daten. Hierzu werden die Ergebnisse der Methoden zunächst mit Hilfe von `pytransform3d` konvertiert und anschließend alle notwendigen Transformationen zur Bewertung berechnet. Danach kann die auf der EN ISO 9283 basierende Metrik zur Bewertung der Genauigkeit verwendet und die Ergebnisse gespeichert werden.

Die verschiedenen erwarteten Maßstäbe der CAD-Modelle und die spezifischen Formate der Ausgaben der verwendeten Methoden zur 6D-Posenbestimmung sind noch einmal in Tabelle 6 zusammengefasst.

Methode	CAD-Maßstab	Ausgabe
MegaPose	mm	Quaternionen $xyzw$ und Translation in m
SAM6D	mm	Rotationsmatrix und Translation in mm
FoundationPose	m	Homogene Transformation 4x4 mit Translation in m

Tabelle 6: Übersicht der erwarteten CAD-Maßstäbe und Ausgabe-Formate der Methoden

Die fertig implementierte Methode, welche in diesem Kapitel beschrieben wurde, soll im weiteren Verlauf verwendet werden, um die implementierten State-of-the-Art-Methoden zur 6D-Posenbestimmung gemäß der in Abschnitt 4.4 beschriebenen Versuchsplanung zu evaluieren.

6. Ergebnisse und Diskussion

In diesem Kapitel werden die zentralen Ergebnisse der durchgeführten Untersuchung präsentiert und kritisch diskutiert. Zu diesem Zweck werden die gewonnenen Daten zunächst gemäß der in Abschnitt 4.4 beschriebenen Versuchsplanung strukturiert dargestellt und anhand relevanter Kriterien analysiert. Hierbei wird die in Abschnitt 4.2 beschriebene Metrik verwendet. Abschließend folgt ein zusammenfassendes Fazit. Ziel ist es, die zentralen Erkenntnisse herauszustellen, ihre wissenschaftliche Relevanz zu bewerten und mögliche zukünftige Forschungsfragen abzuleiten.

Die Ergebnisse der Versuche sind in ausführlicher Form als Tabellen in Anhang A.3 dargestellt, wobei sich die Versuche B3 und C1 aus Versuch A2 ableiten lassen

6.1. Vergleich der Methoden zur 6D-Posenbestimmung

Der Vergleich der drei verwendeten Methoden *MegaPose*, *FoundationPose* und *SAM6D* wird durch die Versuche A1 und A2 der Versuchsplanung abgedeckt. Zum besseren Verständnis und einer objektiven Bewertung der Methoden untereinander werden zunächst nur die Daten des Luxonis-Sensors untersucht. Weiterhin werden nur die Ergebnisse ohne Verdeckung (*base*) und für alle Bewegungen (*all_motions*) betrachtet. Als Auflösung wird 720p verwendet, bei einer Robotergeschwindigkeit von $VEL=100$ sowie einer Bildrate von $FPS=6$.

Der Verlauf der Absolutgenauigkeit AP_p und der Positionswiederholgenauigkeit RP_l für den entsprechenden Versuch ist in Abbildung 69 dargestellt. AP_p und der Positionswiederholgenauigkeit RP_l für den entsprechenden Versuch ist in Abbildung 69 dargestellt.

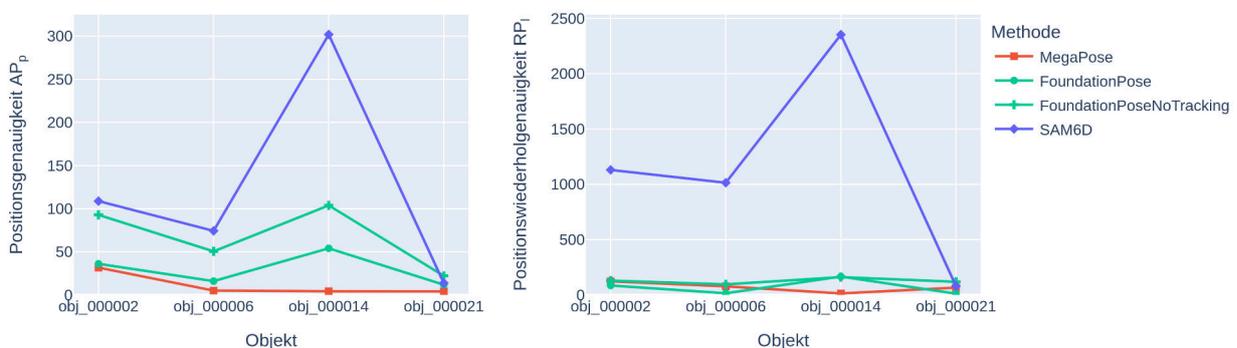


Abbildung 69: Erreichte Genauigkeit AP_p und RP_l der verschiedenen Methoden (720p, Luxonis, base, all_motions, $VEL=100$, $FPS=6$)

Es lässt sich klar erkennen, dass *SAM6D* signifikant weniger genau ist als die Methoden *MegaPose* und *FoundationPose*. Dies liegt daran, dass im Vergleich zu den anderen Methoden die Posenbestimmung mit *SAM6D* nicht auf einer vorgegebenen Region of Interest basiert, welche zwangsläufig korrekt ist, da diese durch die Ground-Truth beschrieben wird. Im Vergleich dazu nutzt *SAM6D* das auf dem SAM-Modell basierende ISM. Dies bedeutet jedoch, dass es zu Fehlannahmen in Bezug auf die Position des Objekts im Bild kommen kann, welche in der Konsequenz zu massiven Abweichungen bei der Posenbestimmung führen können.

Aufgrund dieses Wirkzusammenhangs soll die *SAM6D*-Methode bei der weiteren Betrachtung der Ergebnisse nicht weiter verwendet werden, da kein fairer Vergleich möglich ist.

FoundationPose erreicht, wie erwartet, unter Verwendung des Objekt-Trackings bessere Ergebnisse als ohne (*FoundationPoseNoTracking*). Darüber hinaus ist die Positionsgenauigkeit von *FoundationPose* für die Objekte des HOPE-Datensatzes signifikant geringer als für die Objekte des ITODD-Datensatzes.

MegaPose erreicht auf den Objekten *obj_000006*, *obj_000014* und *obj_000021* Positionsgenauigkeiten von circa 5 mm, jedoch liegt die Positionswiederholungsgenauigkeit weitaus höher, was einer robusten Verwendungsmöglichkeit zunächst widerspricht. Für das Objekt *obj_000021* liegt diese exemplarisch bei 66,41 mm. Im Vergleich dazu kann *FoundationPose* eine Positionswiederholungsgenauigkeit von 11,54 mm erreichen.

Für ein besseres Verständnis sollen zunächst die erreichte Orientierungsgenauigkeit und die Orientierungswiederholungsgenauigkeit der beiden Methoden betrachtet werden, siehe Abbildung 70 und Abbildung 71.

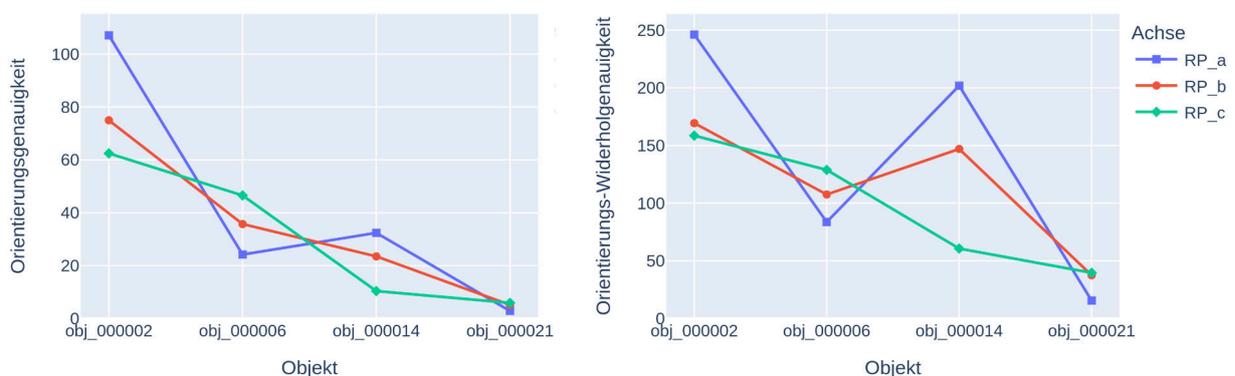


Abbildung 70: Erreichte Orientierungs- und Orientierungs-Wiederholungsgenauigkeit von MegaPose (720p, Luxonis, base, all_motions, VEL=100, FPS=6)

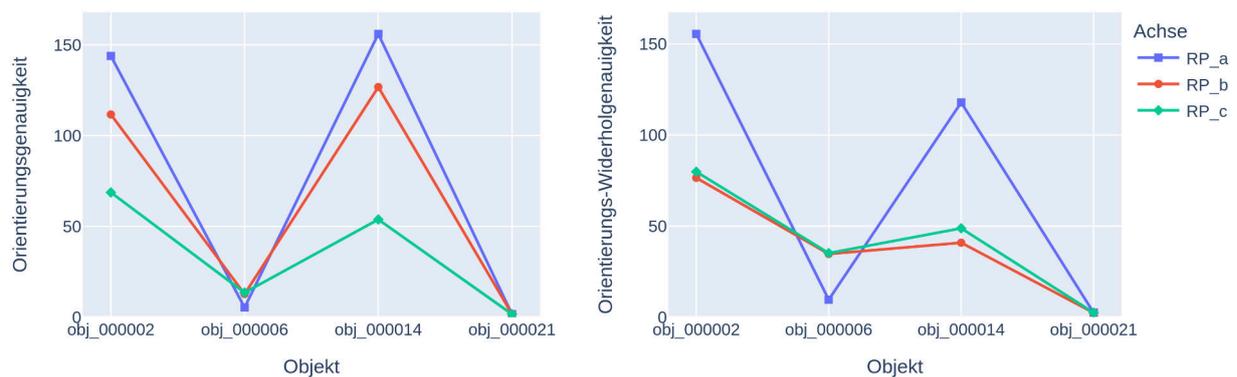


Abbildung 71: Erreichte Orientierungs- und Orientierungs-Wiederholungsgenauigkeit von FoundationPose (720p, Luxonis, base, all_motions, VEL=100, FPS=6)

Wie zu erkennen ist, erreicht *MegaPose* auf *obj_000021* eine bessere Orientierungsgenauigkeit als auf den anderen Objekten. Dies lässt sich erneut durch die stark ausgeprägte Asymmetrie und den größeren Durchmesser im Vergleich zu *obj_000006* erklären. Wie auch bei der Positionsgenauigkeit ist die Orientierungswiederholungsgenauigkeit signifikant geringer und streut daher sehr stark.

Im Vergleich dazu kann *FoundationPose* eine höhere Orientierungswiederholungsgenauigkeit erreichen. Insbesondere für die Objekte des ITODD-Datensatzes sind die erreichten Genauigkeiten allgemein als hervorragend zu bewerten. Wie jedoch auch bei der Positionierung, scheinen die Objekte des HOPE-Datensatzes eine Herausforderung für die Posenbestimmung darzustellen. Auffallend ist weiterhin, dass die Orientierung RP_a , also um die Z-Achse (aus dem Tisch bzw. Template heraus), besonders stark abweicht.

Um die vergleichsweise schlechte Genauigkeit für *obj_000002* und *obj_000014* zu begründen, sollen deren Symmetrieeigenschaften betrachtet werden, siehe dazu Abbildung 72.



Abbildung 72: Beispiel für nicht eindeutige Ansichten der Objekte *obj_000002* und *obj_000014*

Insbesondere für Ansichten, in denen die Objekte horizontal mittig im Bild positioniert sind, ist die Bestimmung der Orientierung um die Z-Achse der Objekte zunehmend unmöglich. Dies liegt daran, dass die Objekttextur nicht mehr klar erkennbar ist, sodass die ansonsten asymmetrische Geometrie der Objekte eine unmögliche Herausforderung für die Posenbestimmung darstellt.

Ein weiterer Grund für dieses Verhalten kann jedoch auch die Nähe der Objekte, insbesondere für `obj_000014`, zum Sensor darstellen, was in den nächsten beiden Abschnitten unter anderem aufgezeigt werden soll.

6.2. Vergleich der Bewegungen

In diesem Abschnitt soll auf die Auswirkung der verschiedenen Bewegungen, vgl. Abschnitt 4.4, auf die erreichbare Genauigkeit eingegangen werden. Die verwendeten Bewegungen `circ_far`, `circ_close` und `zoom` sollen die manuellen Handhabungen von Objekten nachahmen, wobei unterschiedliche Abstände zwischen Sensor und Objekt bestehen, die eine vermutete Auswirkung auf die erreichbare Genauigkeit einer Posenbestimmung haben.

Es soll zunächst Abbildung 73 betrachtet werden, welche die Positions- und Positionswiederholgenauigkeit für die *MegaPose*-Methode entsprechend darstellt.

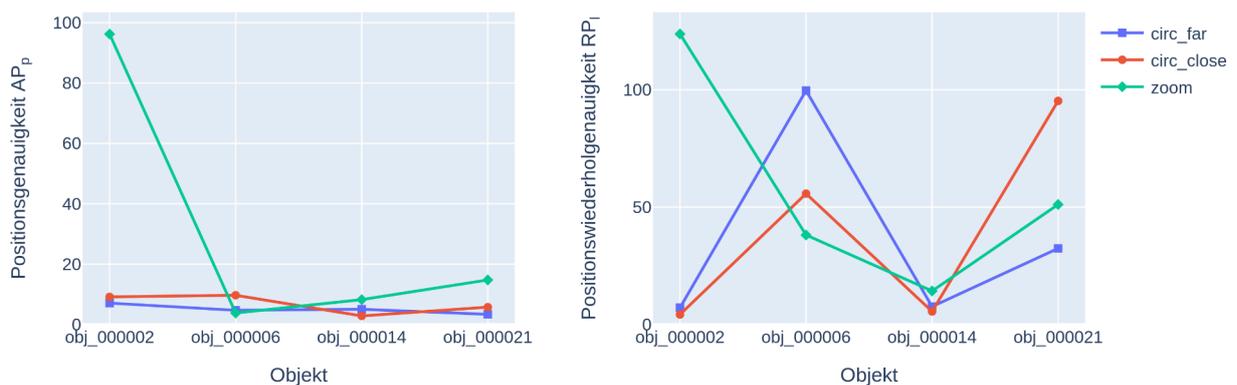


Abbildung 73: Erreichte Genauigkeit AP_P und RP_l nach Bewegung mit MegaPose (720p, Luxonis, base, VEL=100, FPS=6)

Das schlechte Ergebnis für die Zoom-Bewegung bei `obj_000002` lässt sich möglicherweise auch auf die zuvor beschriebene Problematik der Symmetrie zurückführen. Weitere qualifizierte Aussagen lassen sich aufgrund der großen Schwankung insgesamt aus den vorliegenden Daten ableiten.

Die erreichten Genauigkeiten der *FoundationPose*-Methoden sind analog hierzu in Abbildung 74 abgebildet.

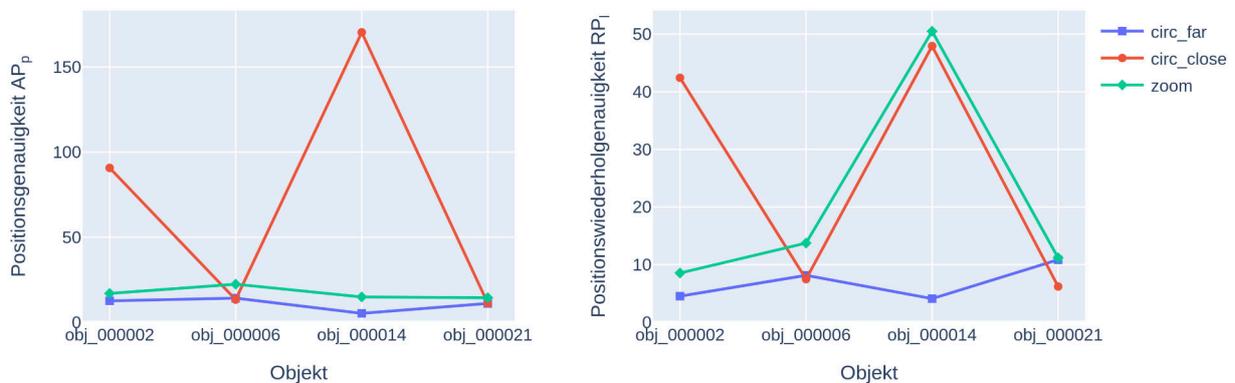


Abbildung 74: Erreichte Genauigkeit AP_P und RP_l nach Bewegung mit *FoundationPose* (720p, Luxonis, base, VEL=100, FPS=6)

Es lässt sich erneut feststellen, dass die teilsymmetrischen Objekte des HOPE-Datensatzes teilweise eine große Herausforderung darstellen. Sehr interessant ist jedoch, dass insbesondere für *obj_000014* die *circ_close*-Bewegung, welche näher an dem Objekt ausgeführt wird, eine besonders signifikante Auswirkung auf die erreichbare Genauigkeit hat. Dies ist umso bemerkenswerter, als dass die Bewegung ansonsten eine nahezu identische Trajektorie wie *circ_far* hat.

Um diesen Zusammenhang zwischen Objektabstand und erreichbarer Genauigkeit näher zu untersuchen, sollen im nächsten Abschnitt die beiden Sensoren miteinander verglichen werden.

6.3. Vergleich der Sensoren

Die Datenaufnahme erfolgte mit zwei verschiedenen Sensoren: der *RealSense D435i* und der *Luxonis OAK-D Pro* (vgl. Abschnitt 2.4.3). Bisher wurden jedoch nur die Aufnahmen des Luxonis-Sensors für die Auswertung der Ergebnisse verwendet. In diesem Abschnitt sollen nun beide Sensoren miteinander verglichen werden. Dies soll insbesondere Erkenntnisse darüber ermöglichen, warum die *FoundationPose*-Methode offensichtliche Schwierigkeiten mit nahen Objekten hat.

Abbildung 75 stellt die erreichte Positions- und Positionswiederholungsgenauigkeit der *MegaPose*-Methode in Abhängigkeit vom Sensor dar. Wie auch zuvor wird dabei eine Auflösung von 720p verwendet, und die Aufnahme wird mit einer Robotergeschwindigkeit von 100 % und einer Bildrate von 6 FPS durchgeführt. Es werden alle Bewegungen ohne Verdeckung betrachtet.

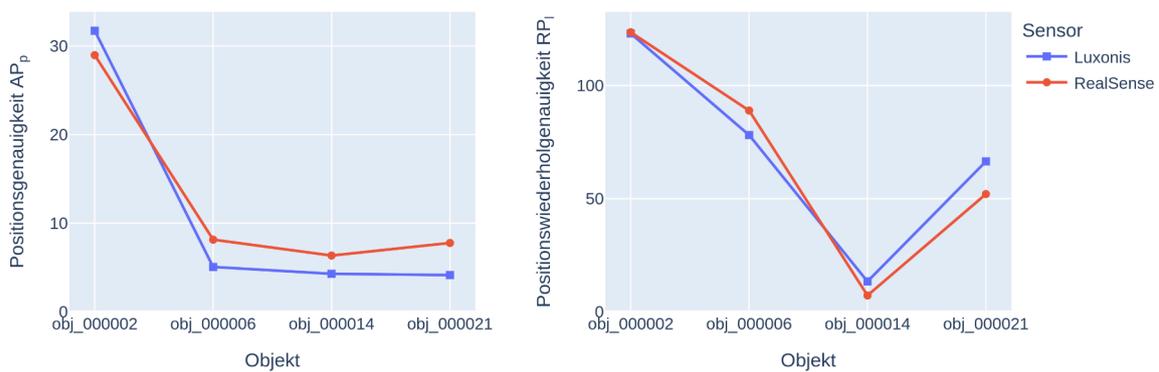


Abbildung 75: Vergleich der erreichten Genauigkeiten AP_P und RP_l nach Sensor mit MegaPose (720p, base, all_motions, vel=100, fps=6)

Der verwendete Sensor hat für *MegaPose* offenbar sehr wenig bis keine Auswirkung auf die erreichbare Genauigkeit. Dies ist insofern plausibel, als *MegaPose* als rein RGB-basierte Methode nur auf die Farbbilder, aber nicht auf die vom Sensor gelieferten Tiefendaten zurückgreift. Beide Sensoren besitzen in dieser Hinsicht sehr ähnliche Eigenschaften.

Im Vergleich hierzu sollen die erreichten Genauigkeiten der *FoundationPose*-Methode in Abhängigkeit vom Sensor in Abbildung 76 betrachtet werden.

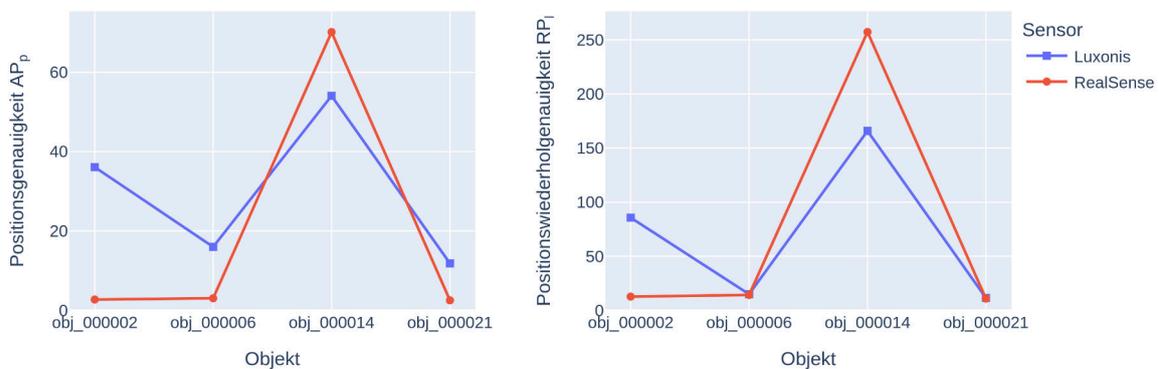


Abbildung 76: Vergleich der erreichten Genauigkeiten AP_P und RP_l nach Sensor mit FoundationPose (720p, base, all_motions, VEL=100, FPS=6)

Wie zu erkennen ist, besteht ein signifikanter Unterschied zwischen den beiden Sensoren, insbesondere für die Objekte des HOPE-Datensatzes. Die erreichte Positionsgenauigkeit des Luxonis-Sensors ist für obj_000002, obj_000006 und obj_000021 geringer als die des RealSense-Sensors. Dies könnte an den subjektiv bewerteten, schlechteren Tiefendaten des Luxonis-Sensors liegen. Hierbei ist jedoch anzumerken, dass der Sensor im Vergleich zum RealSense-Sensor sehr viel umfangreicher konfiguriert werden kann und muss. Eventuell ließe sich dieser Unterschied somit bei einer verbesserten Konfiguration verringern. Für obj_000014 lässt sich das Gegenteil feststellen.

Der Grund hierfür wird in dem geringeren Mindestabstand des Sensors für die Tiefenerkennung vermutet. Während der Luxonis-Sensor Tiefendaten bis zu einem Mindestabstand von 8 cm liefern kann, liegt dieser Abstand für den RealSense-Sensor bei 28 cm. Es ist also naheliegend, dass die geringe Genauigkeit für dieses Objekt hierauf zurückzuführen ist. Dies deckt sich auch mit dem Vergleich der Bewegungen im vorherigen Abschnitt.

6.4. Untersuchung des Einflusses der Sensorauflösung

Die Erkenntnisse aus Abschnitt 4.3 legen die Vermutung nahe, dass eine höhere Sensorauflösung eine genauere Posenbestimmung ermöglicht. Diese Hypothese soll in diesem Abschnitt näher untersucht werden. Hierzu werden die erreichten Positions- und Positionswiederholgenauigkeiten der *MegaPose*- und *FoundationPose*-Methoden bei einer Auflösung von 640x360 Pixeln (360p) sowie 1280x720 Pixeln (720p) miteinander verglichen, siehe Abbildung 77 und Abbildung 78.

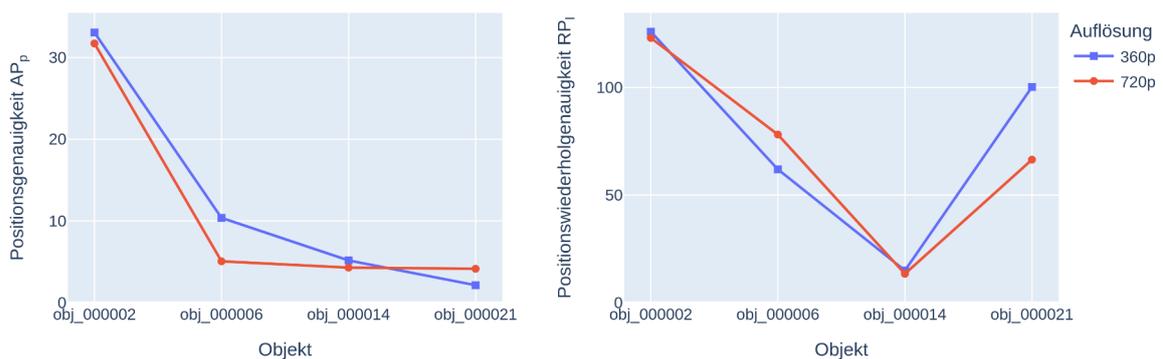


Abbildung 77: Erreichte Genauigkeit AP_P und RP_l nach Auflösung mit MegaPose (Luxonis, base, all_motions, VEL=100, FPS=6)

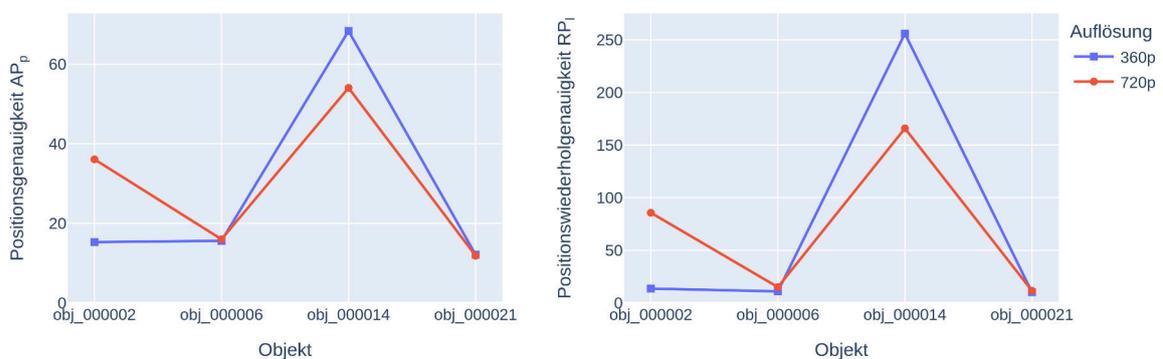


Abbildung 78: Erreichte Genauigkeit AP_P und RP_l nach Auflösung mit FoundationPose (Luxonis, base, all_motions, VEL=100, FPS=6)

Bei der Verwendung der *MegaPose*-Methode hat die Auflösung nur eine geringe Auswirkung auf die erreichte Positionsgenauigkeit. Für `obj_000006` ist diese geringfügig besser bei höherer Auflösung. Für die Positionswiederholungsgenauigkeit lässt sich dies für `obj_000021` ebenso feststellen.

Für *FoundationPose* lassen sich auch keine konkreten Aussagen treffen. Für `obj_000002` scheint eine höhere Auflösung negativ zu sein. Ebenso wie für `obj_000014` kann dieses Verhalten ohne weitere Untersuchungen jedoch nicht qualifiziert beurteilt werden. Dies liegt an den zuvor beschriebenen Herausforderungen, die die Objekte des HOPE-Datensatzes im Versuchsaufbau darstellen.

6.5. Untersuchung des Einflusses der Objektverdeckung

Im Folgenden soll auf den Einfluss der Objektverdeckung eingegangen werden. Wie in Abschnitt 4.4 beschrieben, werden drei Grade der Objektverdeckung für einen Teil der Datenaufnahme verwendet: keine Verdeckung (base), partielle Verdeckung durch eine Hand, die sich dem Objekt annähert (approach), sowie maximale Verdeckung durch vollständiges Umgreifen des Objektes (hand). Die Hypothese hierbei ist, dass ein größeres Maß an Verdeckung die Posenbestimmung zunehmend erschwert. Die *FoundationPose*-Methode sollte hiervon jedoch wesentlich weniger stark beeinträchtigt sein, da sie durch die Verwendung eines Objekt-Trackings zumindest teilweise eine Verdeckung des Objektes kompensieren kann.

Zum Prüfen dieser Hypothese sollen die erreichten Genauigkeiten der Methoden in Abhängigkeit der jeweiligen Objektverdeckung durch die Hand betrachtet werden, siehe Abbildung 79 und Abbildung 80.

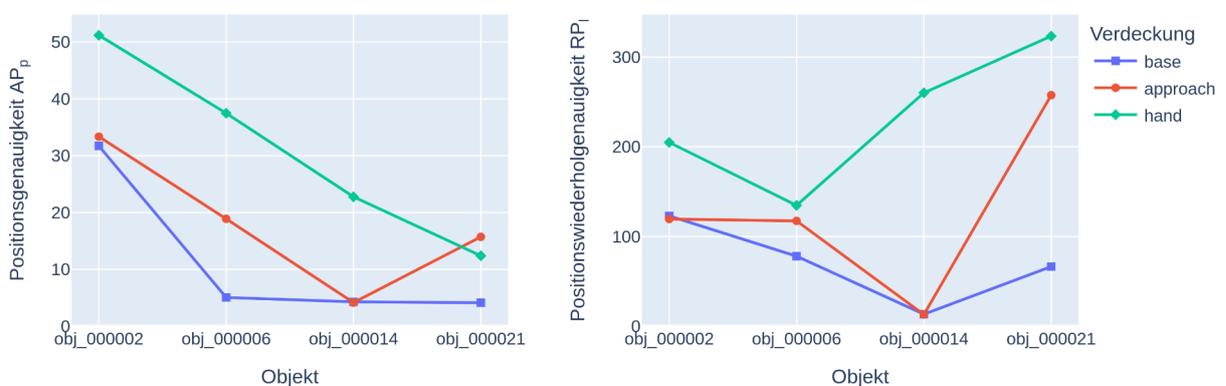


Abbildung 79: Erreichte Genauigkeit AP_p und RP_l nach Verdeckung mit *MegaPose* (720p, Luxonis, all_motions, VEL=100, FPS=6)

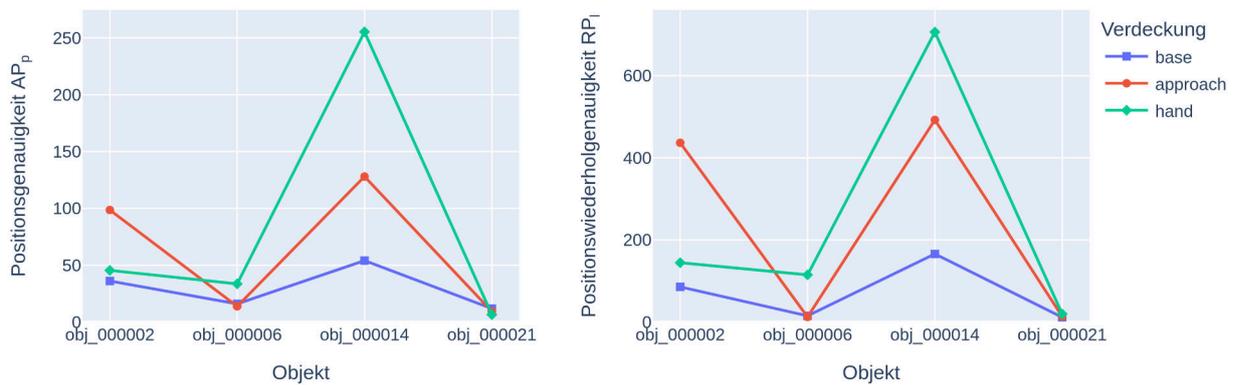


Abbildung 80: Erreichte Genauigkeit AP_P und RP_l nach Verdeckung mit FoundationPose (720p, Luxonis, all_motions, VEL=100, FPS=6)

Die Hypothese lässt sich für *MegaPose* bestätigen, da klar ersichtlich ist, dass die erreichte Positionsgenauigkeit bei höheren Graden an Objektverdeckung abnimmt. obj_000014 ist von der Objektverdeckung weniger stark betroffen, da die Verdeckung aufgrund der Größe des Objektes proportional betrachtet weniger signifikant ist.

Für *FoundationPose* ergibt sich wie erwartet ein anderes Bild. Die Objekte des HOPE-Datensatzes stellen einmal mehr eine Herausforderung dar, während für die Objekte des ITODD-Datensatzes die Auswirkung der Verdeckung jedoch wesentlich weniger stark ausgeprägt ist.

Um die Auswirkung des Objekt-Trackings auf die Genauigkeit während einer Bewegung mit Handverdeckung genauer zu untersuchen, sollen Abbildung 81 und Abbildung 82 betrachtet werden. Diese zeigen die einzelnen Komponenten der Differenz zwischen Soll- und Ist-Pose, also wie stark die jeweiligen Werte zu jedem Schritt der Posenbestimmung voneinander abweichen (vgl. Abbildung 44). Betrachtet wird nur hierbei obj_000021.

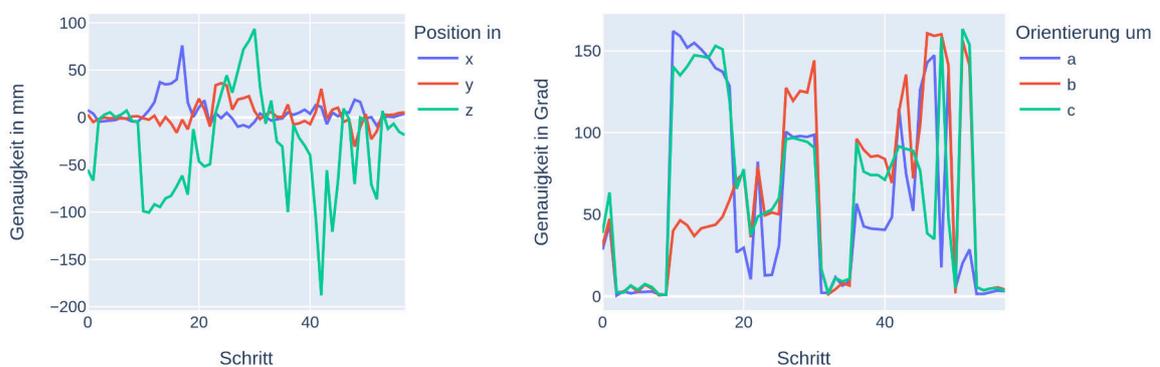


Abbildung 81: Genauigkeit während der circ_far Bewegung mit Handverdeckung mit MegaPose für obj_000021 (720p, Luxonis, VEL=100, FPS=6)

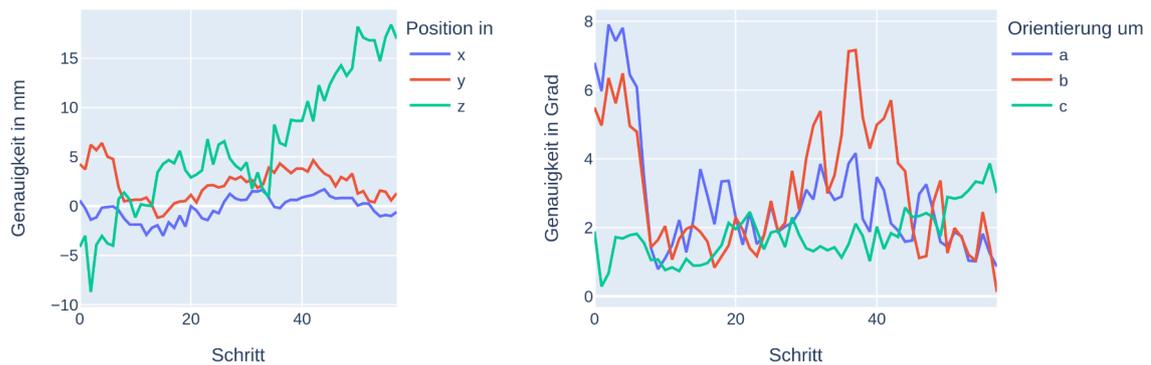


Abbildung 82: Genauigkeit während der *circ_far* Bewegung mit Handverdeckung mit *FoundationPose* für *obj_000021* (720p, Luxonis, VEL=100, FPS=6)

Im Vergleich der beiden Methoden lässt sich der Vorteil des Objekt-Trackings von *FoundationPose* besonders deutlich sehen. Die Genauigkeit der einzelnen Komponenten schwankt wesentlich weniger stark, was sich auch in der sehr viel besseren Positionswiederholgenauigkeit im Vergleich zu *MegaPose* widerspiegelt. Diese beträgt 19,59 mm im Vergleich zu 323,44 mm.

Es kann somit die Aussage getroffen werden, dass Methoden, die über ein inhärentes Objekt-Tracking verfügen, sehr viel robuster gegenüber Objektverdeckung sind. Dennoch hängt die erreichbare Genauigkeit, wie zuvor gezeigt, auch von weiteren Faktoren ab und kann grundsätzlich nicht pauschal gewährleistet werden. In den nächsten beiden Abschnitten sollen weitere mögliche Einflussfaktoren in Bezug auf das Objekt-Tracking betrachtet werden.

6.6. Untersuchung des Einflusses der Bildfrequenz und Robotergeschwindigkeit bei der Aufnahme

Die Bildfrequenz (FPS) und die Robotergeschwindigkeit (Capture VEL), die für die Aufnahme der Daten verwendet werden, haben eine direkte Auswirkung auf den zeitlichen Abstand der einzelnen Frames und somit darauf, wie groß die jeweiligen Veränderungen sind. Es kann die Hypothese aufgestellt werden, dass ein zu großer Abstand kein robustes Objekt-Tracking mehr ermöglicht, da die Veränderungen zu groß sind. Es ist jedoch auch möglich, dass zu geringe Abstände zu einer Verringerung der erreichbaren Genauigkeit führen, da es zu Effekten ähnlich dem Overfitting im Machine Learning kommen kann. Um dies genauer zu untersuchen, soll die *FoundationPose*-Methode in diesem Abschnitt bei Verwendung verschiedener Parameter getestet werden.

Zunächst soll der Einfluss der Bildfrequenz auf die erreichbare Genauigkeit der Posenbestimmung analysiert werden. Hierzu wurden verschiedene Bildfrequenzen (6 FPS, 15 FPS und 30 FPS) unter gleichbleibenden Bedingungen bei einer Robotergeschwindigkeit von 100 % getestet, siehe Abbildung 83.

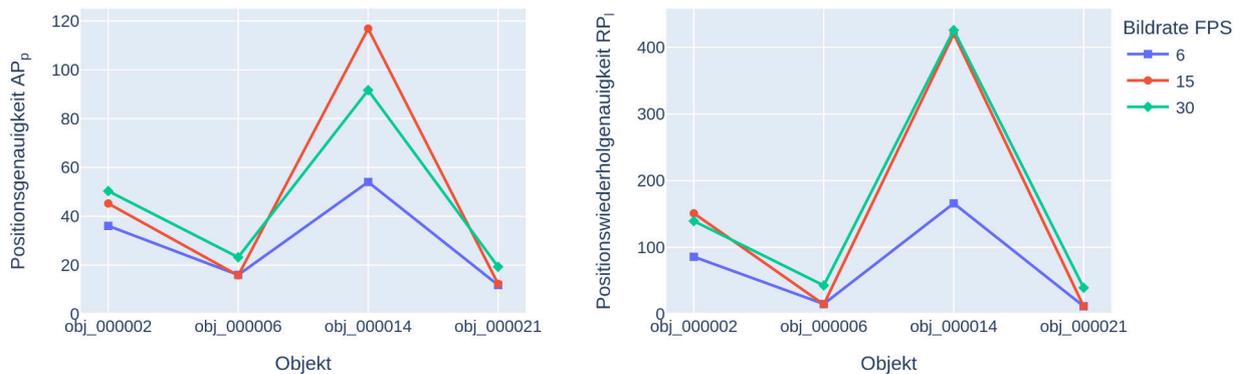


Abbildung 83: Erreichte Genauigkeit AP_P und RP_l nach Bildrate FPS mit FoundationPose (Luxonis, base, all_motions, VEL=100)

Wie zu erkennen ist, nimmt die erreichte Genauigkeit der Posenbestimmung für obj_000014 mit zunehmender Bildrate stark ab. Da dieses Objekt jedoch ohnehin keine objektive Bewertung erlaubt, sollen die Objekte des ITODD-Datensatzes betrachtet werden. Bei diesen ergibt sich keine signifikante Veränderung in Abhängigkeit von der Bildrate.

Im nächsten Schritt wurde der Einfluss der Robotergeschwindigkeit untersucht. Verschiedene Geschwindigkeiten (20 %, 60 % und 100 % der maximalen Geschwindigkeit) wurden bei einer Bildrate von 6 FPS getestet, um zu sehen, wie sich die Geschwindigkeit auf die Genauigkeit der Posenbestimmung auswirkt, siehe Abbildung 84.

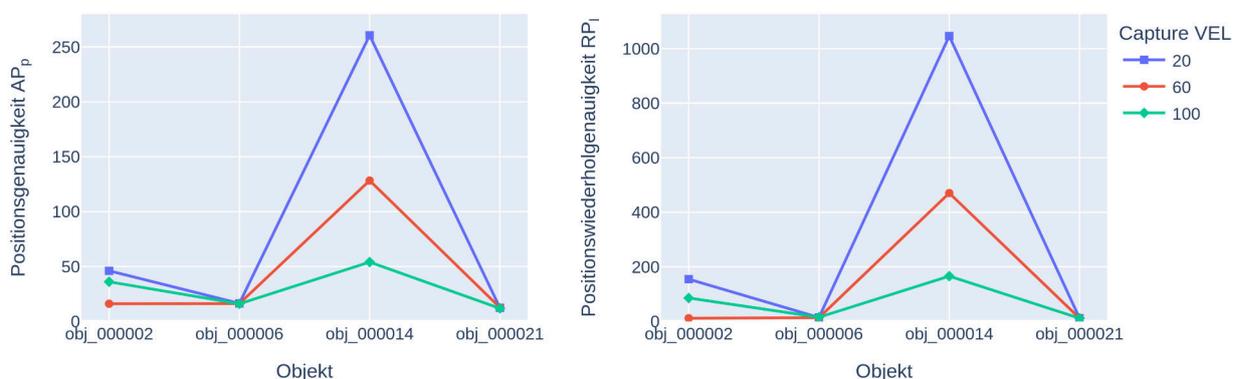


Abbildung 84: Erreichte Genauigkeit AP_P und RP_l nach Robotergeschwindigkeit VEL mit FoundationPose (Luxonis, base, all_motions, FPS=6)

Die Auswirkung der veränderten Robotergeschwindigkeit bei der Aufnahme ist, wie zu erwarten, nahezu identisch mit der Auswirkung der Bildrate. Für die Objekte des ITODD-Datensatzes ergeben sich keine nennenswerten Veränderungen, während für obj_000014 die Genauigkeit mit abnehmender Geschwindigkeit (resultierend in mehr Frames) ebenfalls abnimmt.

Es kann festgehalten werden, dass Bildraten von 6 FPS für die Posenbestimmung mit Objekt-Tracking ausreichend sind. Diese erlauben es außerdem, die Posenbestimmung in Abhängigkeit von der verwendeten Hardware nahezu in Echtzeit auszuführen. Dies bedeutet, dass die Posenbestimmung eines Frames im Tracking schneller ist als die Bildrate.

6.7. Validierung der Genauigkeit bei mehrfacher Wiederholung

In diesem Abschnitt wird die Validierung der Genauigkeit bei mehrfacher Wiederholung anhand einer spezifischen Messreihe untersucht. Ziel ist es, die Reproduzierbarkeit der erzielten Ergebnisse hinsichtlich der erreichten Genauigkeit der Posenbestimmung zu bewerten. Dazu wird der entwickelte Versuchsaufbau mehrfach angewendet und die resultierenden Messdaten systematisch analysiert. Dies ermöglicht eine fundierte Aussage darüber, in welchem Maße der Aufbau konstante und verlässliche Messwerte liefert, wodurch seine Eignung für präzise Anwendungen bestätigt oder infrage gestellt werden kann.

Abbildung 85 zeigt Box-Plots für die Positions- und Positionswiederholgenauigkeit bei fünffacher Wiederholung einer Posenbestimmung mit *FoundationPose*. Hierbei ist anzumerken, dass es sich um fünf separate Datenaufnahmen handelt, d.h. die Objekte wurden jeweils neu auf dem Template platziert.

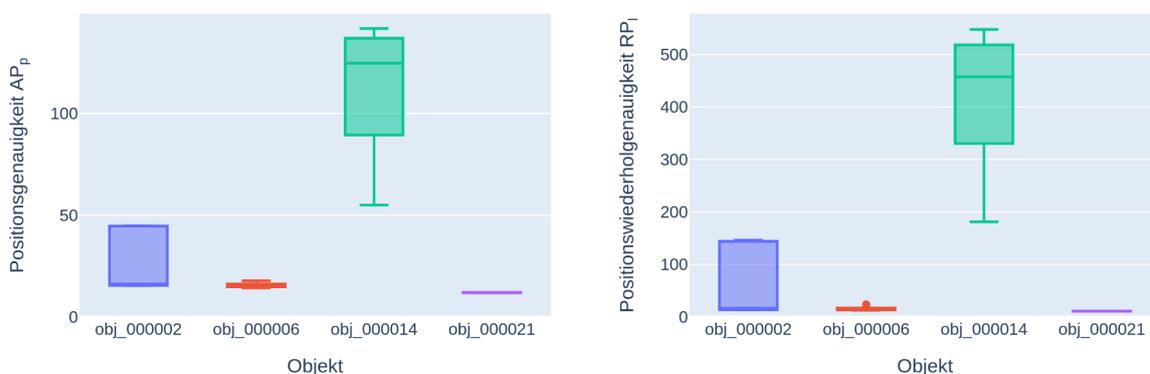


Abbildung 85: Erreichte Genauigkeit AP_p und RP_l bei mehrfacher Wiederholung mit *FoundationPose* (Luxonis, base, all_motions, VEL=100, FPS=6)

Aufgrund der starken Schwankungen bei den Objekten des HOPE-Datensatzes können diese nicht zur Bewertung der Reproduzierbarkeit des entwickelten Verfahrens verwendet werden. Es soll deshalb exemplarisch obj_000021 betrachtet werden, siehe Tabelle 7.

Wdh.	AP _p	ap _x	ap _y	ap _z	ap _a	ap _b	ap _c	RP _l	RP _a	RP _b	RP _c
1	12.07	-0.06	2.39	11.83	1.45	1.7	1.52	10.65	2.4	2.57	2.68
2	11.97	-0.04	2.55	11.69	1.53	1.73	1.56	11.16	2.29	2.58	2.56
3	11.85	-0.07	2.51	11.58	1.34	1.65	1.52	10.61	2.25	2.66	2.6
4	11.82	0.02	2.68	11.52	1.56	1.77	1.61	11.24	2.65	2.74	2.95
5	12.03	-0.01	2.33	11.8	1.61	1.85	1.61	11.09	2.46	2.65	2.88
s	0.1094	0.0391	0.1371	0.1371	0.1038	0.0742	0.0462	0.2976	0.1585	0.0703	0.172

Tabelle 7: Ergebnisse der Bewertungsmetrik für wiederholte Datenaufnahmen und Messungen

Der Wert s beschreibt die Standardabweichung der Genauigkeitswerte, die für eine Stichprobe berechnet wurden. Sämtliche Werte liegen deutlich unter 1 mm bzw. 1°. Dies verdeutlicht, dass sowohl die entwickelte Methode als auch der Versuchsaufbau reproduzierbare Ergebnisse für die Bewertung der 6D-Posenbestimmung liefern können.

6.8. Fazit

Es konnte erfolgreich validiert werden, dass der entwickelte Aufbau zur automatisierten Datenaufnahme zuverlässig funktioniert. Mehrere Messreihen, durchgeführt mit den implementierten Methoden zur 6D-Posenbestimmung, haben bereits erste Erkenntnisse über die Eignung dieser Methoden für den industriellen Einsatz geliefert. Dabei wurde die eigens entwickelte Bewertungsmetrik verwendet.

Die *SAM6D*-Methode bietet zwar den Vorteil, dass keine Region of Interest vorgegeben werden muss, allerdings geschieht dies zu Lasten der erreichbaren Genauigkeit. Ein objektiver Vergleich mit anderen Methoden war deshalb nicht möglich. Die Methoden *MegaPose* und *FoundationPose* konnten ausführlicher untersucht werden. *MegaPose* bietet den Vorteil, dass nur RGB-Daten benötigt werden. Dadurch ist die Methode unter anderem robuster bei der Erkennung von Objekten, die mit einem kleinen Abstand zum Sensor aufgenommen werden. *FoundationPose* bietet hingegen den Vorteil, dass die Posenbestimmung von Objekten teilweise auch bei Verdeckung möglich ist, da die Methode Objekt-Tracking nutzt.

Darüber hinaus konnte demonstriert werden, dass der Aufbau reproduzierbare Ergebnisse liefert und somit zur systematischen Evaluierung der 6D-Posenbestimmungen verwendet werden kann.

7. Zusammenfassung und Ausblick

Das Ziel dieser wissenschaftlichen Arbeit bestand darin, eine Methodik zur automatisierten Datenerfassung für die Evaluationszwecke der 6D-Posenbestimmung zu entwickeln und aktuelle State-of-the-Art-Methoden zur Posenbestimmung neuartiger starrer Objekte quantitativ zu bewerten. Hierbei kam ein Industrieroboter zum Einsatz, dessen hohe Genauigkeit es in Verbindung mit einer template-basierten Methode ermöglichte, die erfassten Daten mit einer präzisen Ground-Truth zu annotieren.

Die im Rahmen dieser Arbeit implementierten Methoden und die entwickelte Bewertungsmetrik ermöglichten eine umfassende Evaluation. Die Ergebnisse zeigten, dass die Methoden unterschiedliche Stärken und Schwächen aufweisen. Insbesondere stellten einige Objekte und spezifische Anwendungsfälle nach wie vor signifikante Herausforderungen dar, was darauf hinweist, dass für den robusten und flexiblen Einsatz in industriellen Anwendungen weitere Verbesserungen notwendig sind. Während die Methoden im Allgemeinen eine gute Absolutgenauigkeit der Pose zeigen, weisen sie eine beträchtliche Streuung auf, was sich negativ auf die Wiederholgenauigkeit auswirkt.

Mit dem entwickelten Verfahren kann die Eignung verschiedener Methoden für spezifische Anwendungsfälle in zukünftigen, ausführlicheren Versuchen weiter untersucht werden. Darüber hinaus kann der entwickelte Aufbau zur Erstellung umfangreicher und exakt annotierter Testdatensätze genutzt werden. Dies schließt die bestehende Lücke in Bezug auf spezifische Anwendungsbereiche und treibt die Weiterentwicklung robuster und flexibler Methoden zur 6D-Posenbestimmung voran.

Zusammenfassend leistet diese Arbeit einen wichtigen Beitrag zur Bewertung und Optimierung der 6D-Posenbestimmung und legt eine solide Grundlage für weitere Forschungsprojekte sowie praktische Anwendungen in der industriellen Automatisierung.

Aus den im Rahmen dieser Arbeit gewonnenen Erkenntnissen und Methoden lassen sich diverse Aufgaben für die zukünftige Forschung ableiten (siehe Abbildung 86).



Abbildung 86: Ausblick auf zukünftige Forschung

Eine umfassende Evaluierung mit mehreren Objekten kann durchgeführt werden, um Erkenntnisse über deren Eignung für spezifische Aufgabenstellungen zu gewinnen. Die Implementierung der Methoden auf Cluster-Ressourcen erscheint hierfür als sinnvoll.

Zur präzisen Annotation der Daten mit Handverdeckungen können Segmentierungsmodelle wie SAM 2 verwendet werden. Dadurch lässt sich das Ausmaß der Verdeckung exakt bestimmen und für evaluale Zwecke nutzen.

Spezifische Datensätze, insbesondere mit einem Fokus auf die industrielle Produktion, die Hand-Objekt-Interaktionen und/oder Roboter-Objekt-Interaktionen beinhalten, können mit der entwickelten Methode erfasst werden. Dies trägt dazu bei, die Forschungslücke in diesem Bereich weiter zu schließen.

Die Methoden zur 6D-Posenbestimmung können auf aktueller Hardware, wie der Apple Vision Pro oder der Microsoft HoloLens 2, implementiert werden. Dabei kann die Vorgabe der Region of Interest durch eine Point-and-Click-Interaktion des Benutzers erfolgen. Anschließend kann eine Integration in das Ökosystem des vorgestellten HRC-Hubs erfolgen, um das Potenzial der Programmierung durch Vormachen auch kleinen und mittleren Unternehmen zugänglich zu machen.

Als weitere Verbesserung für Methoden zur Posenbestimmung können Ansätze entwickelt werden, die die Bestimmung skalensvariabler Bauteile ermöglichen oder auf Multi-View-Ansätze zurückgreifen, um höhere Genauigkeiten zu erzielen und robuster gegenüber Verdeckungen zu werden.



8. Literaturverzeichnis

- [1] Torge Mattis Kolditz. *Methoden zur Steigerung der Flexibilität und Robustheit einer aerodynamischen Zuführanlage für die automatisierte Montage*. Berichte aus dem match 2024, Band 1. Garbsen: TEWISS Verlag, 2024. 125 S. ISBN: 978-3-95900-931-7.
- [2] Zsolt Kemény u. a. „Human–Robot Collaboration in Manufacturing: A Multi-agent View“. In: *Advanced Human-Robot Collaboration in Manufacturing*. Hrsg. von Lihui Wang u. a. Cham: Springer International Publishing, 2021, S. 3–41. ISBN: 978-3-030-69178-3. DOI: 10.1007/978-3-030-69178-3_1. URL: https://doi.org/10.1007/978-3-030-69178-3_1 (besucht am 25.07.2024).
- [3] Sebastian Blankemeyer, David Wendorff und Annika Raatz. „A Hand-Interaction Model for Augmented Reality Enhanced Human-Robot Collaboration“. In: *CIRP Annals* 73.1 (2024), S. 17–20. ISSN: 0007-8506. DOI: 10.1016/j.cirp.2024.04.001. URL: <https://www.sciencedirect.com/science/article/pii/S0007850624000131>.
- [4] Azemeraw Tadesse Mengistu und Roberto Panizzolo. „Metrics for Measuring Industrial Sustainability Performance in Small and Medium-Sized Enterprises“. In: *International Journal of Productivity and Performance Management* 73.11 (1. Jan. 2023), S. 46–68. ISSN: 1741-0401. DOI: 10.1108/IJPPM-04-2022-0200. URL: <https://doi.org/10.1108/IJPPM-04-2022-0200> (besucht am 01.08.2024).
- [5] Stefan Koos. „The German Supply Chain Due Diligence Act 2021 and Its Impact on Globally Operating German Companies“. In: 2nd Riau Annual Meeting on Law and Social Sciences (RAMLAS 2021). Atlantis Press, 23. Apr. 2022, S. 111–115. ISBN: 978-94-6239-569-5. DOI: 10.2991/assehr.k.220406.027. URL: <https://www.atlantis-press.com/proceedings/ramlas-21/125973359> (besucht am 31.07.2024).
- [6] *Directive - EU - 2024/1760 - EN - EUR-Lex*. URL: <https://eur-lex.europa.eu/eli/dir/2024/1760/oj> (besucht am 31.07.2024).
- [7] Rainer Müller u. a., Hrsg. *Handbuch Mensch-Roboter-Kollaboration*. 2., aktualisierte Auflage. München: Hanser, 2024. 485 S. ISBN: 978-3-446-47411-6.
- [8] Lihui Wang u. a., Hrsg. *Advanced Human-Robot Collaboration in Manufacturing*. Cham: Springer International Publishing, 2021. ISBN: 978-3-030-69177-6 978-3-030-69178-3. DOI: 10.1007/978-3-030-69178-3. URL: <https://link.springer.com/10.1007/978-3-030-69178-3> (besucht am 12.01.2024).

- [9] Gregory F. Rossano u. a. „Easy Robot Programming Concepts: An Industrial Perspective“. In: *2013 IEEE International Conference on Automation Science and Engineering (CASE)*. 2013 IEEE International Conference on Automation Science and Engineering (CASE). Aug. 2013, S. 1119–1126. DOI: 10.1109/CoASE.2013.6654035. URL: <https://ieeexplore.ieee.org/document/6654035> (besucht am 31. 07. 2024).
- [10] Peter George u. a. „Task Complexity and the Skills Dilemma in the Programming and Control of Collaborative Robots for Manufacturing“. In: *Applied Sciences* 13.7 (6. Apr. 2023), S. 4635. ISSN: 2076-3417. DOI: 10.3390/app13074635. URL: <https://www.mdpi.com/2076-3417/13/7/4635> (besucht am 11. 12. 2023).
- [11] Gopika Ajaykumar und Chien-Ming Huang. *Multimodal Robot Programming by Demonstration: A Preliminary Exploration*. 17. Jan. 2023. DOI: 10.48550/arXiv.2301.07189. arXiv: 2301.07189 [cs]. URL: <http://arxiv.org/abs/2301.07189> (besucht am 31. 07. 2024). Vorveröffentlichung.
- [12] Mohammad Ghahramani, Aleksandar Vakanski und Farrokh Janabi-Sharifi. „6D Object Pose Estimation for Robot Programming by Demonstration“. In: *Progress in Optomechatronic Technologies*. Hrsg. von Amalia Martínez-García u. a. Singapore: Springer, 2019, S. 93–101. ISBN: 978-981-329-632-9. DOI: 10.1007/978-981-32-9632-9_11.
- [13] Jian Guan u. a. „A Survey of 6DoF Object Pose Estimation Methods for Different Application Scenarios“. In: *Sensors* 24.4 (7. Feb. 2024), S. 1076. ISSN: 1424-8220. DOI: 10.3390/s24041076. URL: <https://www.mdpi.com/1424-8220/24/4/1076> (besucht am 25. 04. 2024).
- [14] Tomas Hodan u. a. *BOP Challenge 2023 on Detection, Segmentation and Pose Estimation of Seen and Unseen Rigid Objects*. 16. Apr. 2024. arXiv: 2403.09799 [cs]. URL: <http://arxiv.org/abs/2403.09799> (besucht am 22. 04. 2024). Vorveröffentlichung.
- [15] Robert X. Gao u. a. „Human Motion Recognition and Prediction for Robot Control“. In: *Advanced Human-Robot Collaboration in Manufacturing*. Hrsg. von Lihui Wang u. a. Cham: Springer International Publishing, 2021, S. 261–282. ISBN: 978-3-030-69178-3. DOI: 10.1007/978-3-030-69178-3_11. URL: https://doi.org/10.1007/978-3-030-69178-3_11 (besucht am 18. 07. 2024).
- [16] *DIN EN ISO 10218-1:2021-09, Robotik_ - Sicherheitsanforderungen_ - Teil_1: Industrieroboter (ISO/DIS_10218-1.2:2021); Deutsche und Englische Fassung prEN_ISO_10218-1:2021*. 2021. DOI: 10.31030/3272912. URL: <https://www.beuth.de/de/-/-/341406648> (besucht am 03. 08. 2024).

- [17] Ryo Suzuki u. a. „Augmented Reality and Robotics: A Survey and Taxonomy for AR-enhanced Human-Robot Interaction and Robotic Interfaces“. In: *CHI Conference on Human Factors in Computing Systems*. CHI '22: CHI Conference on Human Factors in Computing Systems. New Orleans LA USA: ACM, 29. Apr. 2022, S. 1–33. ISBN: 978-1-4503-9157-3. DOI: 10.1145/3491102.3517719. URL: <https://dl.acm.org/doi/10.1145/3491102.3517719> (besucht am 11. 12. 2023).
- [18] Zhanat Makhataeva und Huseyin Varol. „Augmented Reality for Robotics: A Review“. In: *Robotics* 9.2 (2. Apr. 2020), S. 21. ISSN: 2218-6581. DOI: 10.3390/robotics9020021. URL: <https://www.mdpi.com/2218-6581/9/2/21> (besucht am 11. 12. 2023).
- [19] Morteza Dianatfar, Jyrki Latokartano und Minna Lanz. „Review on Existing VR/AR Solutions in Human–Robot Collaboration“. In: *Procedia CIRP* 97 (2021), S. 407–411. ISSN: 22128271. DOI: 10.1016/j.procir.2020.05.259. URL: <https://linkinghub.elsevier.com/retrieve/pii/S2212827120314815> (besucht am 04. 12. 2023).
- [20] Wei Fang u. a. „Head-Mounted Display Augmented Reality in Manufacturing: A Systematic Review“. In: *Robotics and Computer-Integrated Manufacturing* 83 (Okt. 2023), S. 102567. ISSN: 07365845. DOI: 10.1016/j.rcim.2023.102567. URL: <https://linkinghub.elsevier.com/retrieve/pii/S0736584523000431> (besucht am 11. 12. 2023).
- [21] Kerstin Fischer u. a. „A Comparison of Types of Robot Control for Programming by Demonstration“. In: *2016 11th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. 2016 11th ACM/IEEE International Conference on Human-Robot Interaction (HRI). März 2016, S. 213–220. DOI: 10.1109/HRI.2016.7451754. URL: <https://ieeexplore.ieee.org/document/7451754> (besucht am 03. 08. 2024).
- [22] Brenna D. Argall u. a. „A Survey of Robot Learning from Demonstration“. In: *Robotics and Autonomous Systems* 57.5 (31. Mai 2009), S. 469–483. ISSN: 0921-8890. DOI: 10.1016/j.robot.2008.10.024. URL: <https://www.sciencedirect.com/science/article/pii/S0921889008001772> (besucht am 22. 07. 2024).
- [23] Hongyi Liu und Lihui Wang. „Latest Developments of Gesture Recognition for Human–Robot Collaboration“. In: *Advanced Human-Robot Collaboration in Manufacturing*. Hrsg. von Lihui Wang u. a. Cham: Springer International Publishing, 2021, S. 43–68. ISBN: 978-3-030-69178-3. DOI: 10.1007/978-3-030-69178-3_2. URL: https://doi.org/10.1007/978-3-030-69178-3_2 (besucht am 18. 07. 2024).
- [24] Jiafan Zhang, Yue Wang und Rong Xiong. „Industrial Robot Programming by Demonstration“. In: *2016 International Conference on Advanced Robotics and Mechatronics (ICARM)*. 2016 International Conference on Advanced Robotics and Mechatronics (ICARM). Macau, China: IEEE, Aug. 2016, S. 300–305. ISBN: 978-1-5090-3364-5. DOI: 10.1109/ICARM.

- 2016.7606936. URL: <http://ieeexplore.ieee.org/document/7606936/> (besucht am 25.03.2024).
- [25] Wenchao Zou und Mayur Andulkar. „Development of Robot Programming System through the Use of Augmented Reality for Assembly Tasks“. In: (2018).
- [26] Sebastian Blankemeyer u. a. „Intuitive Robot Programming Using Augmented Reality“. In: *Procedia CIRP* 76 (2018), S. 155–160. ISSN: 22128271. DOI: 10.1016/j.procir.2018.02.028. URL: <https://linkinghub.elsevier.com/retrieve/pii/S2212827118300933> (besucht am 11.03.2024).
- [27] Sotiris Aivaliotis u. a. „An Augmented Reality Software Suite Enabling Seamless Human Robot Interaction“. In: *International Journal of Computer Integrated Manufacturing* 36.1 (2. Jan. 2023), S. 3–29. ISSN: 0951-192X, 1362-3052. DOI: 10.1080/0951192X.2022.2104459. URL: <https://www.tandfonline.com/doi/full/10.1080/0951192X.2022.2104459> (besucht am 05.12.2023).
- [28] Ajay Mandlekar u. a. *Human-in-the-Loop Task and Motion Planning for Imitation Learning*. 24. Okt. 2023. DOI: 10.48550/arXiv.2310.16014. arXiv: 2310.16014 [cs]. URL: <http://arxiv.org/abs/2310.16014> (besucht am 01.08.2024). Vorveröffentlichung.
- [29] Bingjie Tang u. a. *AutoMate: Specialist and Generalist Assembly Policies over Diverse Geometries*. 10. Juli 2024. arXiv: 2407.08028 [cs]. URL: <http://arxiv.org/abs/2407.08028> (besucht am 01.08.2024). Vorveröffentlichung.
- [30] Tomáš Hodaň. „Pose Estimation of Specific Rigid Objects“. Diss. Prague, 2021.
- [31] Yann Labbé. „Pose Estimation of Rigid Objects and Robots“. Diss. Ecole Normale Supérieure, 5. Juni 2023. URL: <https://hal.science/tel-04124865> (besucht am 18.07.2024).
- [32] Tomáš Hodaň, Jiří Matas und Štěpán Obdržálek. „On Evaluation of 6D Object Pose Estimation“. In: *Computer Vision – ECCV 2016 Workshops*. Hrsg. von Gang Hua und Hervé Jégou. Bd. 9915. Cham: Springer International Publishing, 2016, S. 606–619. ISBN: 978-3-319-49408-1 978-3-319-49409-8. DOI: 10.1007/978-3-319-49409-8_52. URL: http://link.springer.com/10.1007/978-3-319-49409-8_52 (besucht am 05.02.2024).
- [33] Stefan Hinterstoisser u. a. „Model Based Training, Detection and Pose Estimation of Texture-Less 3D Objects in Heavily Cluttered Scenes“. In: *Computer Vision – ACCV 2012*. Hrsg. von Kyoung Mu Lee u. a. Bearb. von David Hutchison u. a. Bd. 7724. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, S. 548–562. ISBN: 978-3-642-37330-5 978-3-642-37331-2. DOI: 10.1007/978-3-642-37331-2_42. URL: http://link.springer.com/10.1007/978-3-642-37331-2_42 (besucht am 18.05.2024).

- [34] Adam Morawiec. *Orientations and Rotations*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004. ISBN: 978-3-642-07386-1 978-3-662-09156-2. DOI: 10.1007/978-3-662-09156-2. URL: <http://link.springer.com/10.1007/978-3-662-09156-2> (besucht am 29.06.2024).
- [35] Yu Xiang u. a. *PoseCNN: A Convolutional Neural Network for 6D Object Pose Estimation in Cluttered Scenes*. 26. Mai 2018. arXiv: 1711.00199 [cs]. URL: <http://arxiv.org/abs/1711.00199> (besucht am 17.04.2024). Vorveröffentlichung.
- [36] *BOP: Benchmark for 6D Object Pose Estimation*. URL: <https://bop.felk.cvut.cz/> (besucht am 29.06.2024).
- [37] Tomáš Hodaň u. a. „BOP: Benchmark for 6D Object Pose Estimation“. In: *Computer Vision – ECCV 2018*. Hrsg. von Vittorio Ferrari u. a. Bd. 11214. Cham: Springer International Publishing, 2018, S. 19–35. ISBN: 978-3-030-01248-9 978-3-030-01249-6. DOI: 10.1007/978-3-030-01249-6_2. URL: https://link.springer.com/10.1007/978-3-030-01249-6_2 (besucht am 05.02.2024).
- [38] Tomas Hodan u. a. *BOP Challenge 2020 on 6D Object Localization*. 13. Okt. 2020. arXiv: 2009.07378 [cs]. URL: <http://arxiv.org/abs/2009.07378> (besucht am 12.02.2024). Vorveröffentlichung.
- [39] Martin Sundermeyer u. a. *BOP Challenge 2022 on Detection, Segmentation and Pose Estimation of Specific Rigid Objects*. 25. Feb. 2023. arXiv: 2302.13075 [cs]. URL: <http://arxiv.org/abs/2302.13075> (besucht am 12.02.2024). Vorveröffentlichung.
- [40] *BOP: Benchmark for 6D Object Pose Estimation*. URL: <https://bop.felk.cvut.cz/challenges/bop-challenge-2024/> (besucht am 26.07.2024).
- [41] Prithviraj Banerjee u. a. *Introducing HOT3D: An Egocentric Dataset for 3D Hand and Object Tracking*. 13. Juni 2024. DOI: 10.48550/arXiv.2406.09598. arXiv: 2406.09598 [cs]. URL: <http://arxiv.org/abs/2406.09598> (besucht am 22.07.2024). Vorveröffentlichung.
- [42] Stephen Tyree u. a. *6-DoF Pose Estimation of Household Objects for Robotic Manipulation: An Accessible Dataset and Benchmark*. 15. Dez. 2022. DOI: 10.48550/arXiv.2203.05701. arXiv: 2203.05701 [cs]. URL: <http://arxiv.org/abs/2203.05701> (besucht am 26.07.2024). Vorveröffentlichung.
- [43] Andrew Guo u. a. *HANDAL: A Dataset of Real-World Manipulable Object Categories with Pose Annotations, Affordances, and Reconstructions*. 2. Aug. 2023. DOI: 10.48550/arXiv.2308.01477. arXiv: 2308.01477 [cs]. URL: <http://arxiv.org/abs/2308.01477> (besucht am 05.08.2024). Vorveröffentlichung.

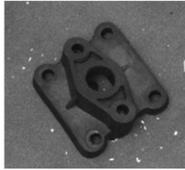
- [44] Berk Calli u. a. „The YCB Object and Model Set: Towards Common Benchmarks for Manipulation Research“. In: *2015 International Conference on Advanced Robotics (ICAR)*. 2015 International Conference on Advanced Robotics (ICAR). Juli 2015, S. 510–517. DOI: 10.1109/ICAR.2015.7251504. URL: <https://ieeexplore.ieee.org/document/7251504> (besucht am 05.08.2024).
- [45] Tomáš Hodaň u. a. „T-LESS: An RGB-D Dataset for 6D Pose Estimation of Texture-Less Objects“. In: *IEEE Winter Conference on Applications of Computer Vision (WACV) (2017)*.
- [46] Bertram Drost u. a. „Introducing MVTec ITODD — A Dataset for 3D Object Recognition in Industry“. In: *2017 IEEE International Conference on Computer Vision Workshops (ICCVW)*. 2017 IEEE International Conference on Computer Vision Workshop (ICCVW). Venice, Italy: IEEE, Okt. 2017, S. 2200–2208. ISBN: 978-1-5386-1034-3. DOI: 10.1109/ICCVW.2017.257. URL: <http://ieeexplore.ieee.org/document/8265467/> (besucht am 22.12.2023).
- [47] *OAK-D Pro — DepthAI Hardware Documentation 1.0.0 Documentation*. URL: <https://docs.luxonis.com/projects/hardware/en/latest/pages/DM9098pro/> (besucht am 22.04.2024).
- [48] *Intel RealSense D400 Series Product Family Datasheet*. Intel® RealSense™ Developer Documentation. URL: <https://dev.intelrealsense.com/docs/intel-realsense-d400-series-product-family-datasheet> (besucht am 22.04.2024).
- [49] *Robotics Vision Core 2*. Luxonis. URL: <https://docs.luxonis.com> (besucht am 05.08.2024).
- [50] Yongzhi Su u. a. *ZebraPose: Coarse to Fine Surface Encoding for 6DoF Object Pose Estimation*. 29. März 2022. arXiv: 2203.09418 [cs]. URL: <http://arxiv.org/abs/2203.09418> (besucht am 21.12.2023). Vorveröffentlichung.
- [51] Yann Labbé u. a. *CosyPose: Consistent Multi-View Multi-Object 6D Pose Estimation*. 19. Aug. 2020. arXiv: 2008.08465 [cs]. URL: <http://arxiv.org/abs/2008.08465> (besucht am 20.12.2023). Vorveröffentlichung.
- [52] Yann Labbé u. a. *MegaPose: 6D Pose Estimation of Novel Objects via Render & Compare*. 13. Dez. 2022. arXiv: 2212.06870 [cs]. URL: <http://arxiv.org/abs/2212.06870> (besucht am 11.12.2023). Vorveröffentlichung.
- [53] Evin Pinar Örnek u. a. *FoundPose: Unseen Object Pose Estimation with Foundation Features*. 30. Nov. 2023. arXiv: 2311.18809 [cs]. URL: <http://arxiv.org/abs/2311.18809> (besucht am 21.12.2023). Vorveröffentlichung.

- [54] Jiehong Lin u. a. *SAM-6D: Segment Anything Model Meets Zero-Shot 6D Object Pose Estimation*. 6. März 2024. arXiv: 2311.15707 [cs]. URL: <http://arxiv.org/abs/2311.15707> (besucht am 25.03.2024). Vorveröffentlichung.
- [55] Alexander Kirillov u. a. „Segment Anything“. Version 1. In: (2023). DOI: 10.48550/ARXIV.2304.02643. URL: <https://arxiv.org/abs/2304.02643> (besucht am 25.03.2024).
- [56] Charles R. Qi u. a. *PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation*. 10. Apr. 2017. arXiv: 1612.00593 [cs]. URL: <http://arxiv.org/abs/1612.00593> (besucht am 22.04.2024). Vorveröffentlichung.
- [57] Charles R. Qi u. a. *PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space*. 7. Juni 2017. arXiv: 1706.02413 [cs]. URL: <http://arxiv.org/abs/1706.02413> (besucht am 29.07.2024). Vorveröffentlichung.
- [58] Nikhila Ravi u. a. „SAM 2: Segment Anything in Images and Videos“. In: ().
- [59] Bowen Wen u. a. *FoundationPose: Unified 6D Pose Estimation and Tracking of Novel Objects*. 26. März 2024. arXiv: 2312.08344 [cs]. URL: <http://arxiv.org/abs/2312.08344> (besucht am 15.04.2024). Vorveröffentlichung.
- [60] Bowen Wen u. a. „BundleSDF: Neural 6-DoF Tracking and 3D Reconstruction of Unknown Objects“. In: *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Vancouver, BC, Canada: IEEE, Juni 2023, S. 606–617. ISBN: 9798350301298. DOI: 10.1109/CVPR52729.2023.00066. URL: <https://ieeexplore.ieee.org/document/10203995/> (besucht am 20.12.2023).
- [61] Ashish Vaswani u. a. *Attention Is All You Need*. 1. Aug. 2023. arXiv: 1706.03762 [cs]. URL: <http://arxiv.org/abs/1706.03762> (besucht am 17.04.2024). Vorveröffentlichung.
- [62] *OpenCV: Camera Calibration and 3D Reconstruction*. URL: https://docs.opencv.org/4.10.0/d9/d0c/group__calib3d.html (besucht am 24.07.2024).
- [63] Richard Hartley und Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Second edition. Cambridge, UK: Cambridge University Press, 2004. ISBN: 978-0-511-18711-7.
- [64] Christian Demant, Bernd Streicher-Abel und Axel Springhoff. *Industrielle Bildverarbeitung: Wie optische Qualitätskontrolle wirklich funktioniert*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011. ISBN: 978-3-642-13096-0 978-3-642-13097-7. DOI: 10.1007/978-3-642-13097-7. URL: <https://link.springer.com/10.1007/978-3-642-13097-7> (besucht am 22.04.2024).
- [65] *DIN EN ISO 9283*. DIN Deutsches Institut für Normung e. V., 1999.

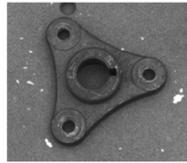
- [66] G. Berndt, E. Hultsch und H. Weinhold. „Fertigungstoleranz Und Funktionstoleranz“. In: *Technische Rundschau, Bern* 61.28 (1969), S. 15. ISSN: 0040-148X.
- [67] KUKA Deutschland GmbH. *LBR Iiwa 14 R820*. 12. Aug. 2021.
- [68] KUKA Deutschland GmbH. *KUKA Sunrise.OS 1.15 - Operating Instructions for End Users*. 24. Jan. 2018.
- [69] Alexander Fabisch. „Pytransform3d: 3D Transformations for Python“. In: *Journal of Open Source Software* 4.33 (31. Jan. 2019), S. 1159. ISSN: 2475-9066. DOI: 10.21105/joss.01159. URL: <https://joss.theoj.org/papers/10.21105/joss.01159> (besucht am 29.04.2024).
- [70] *OpenCV: Pose Estimation*. URL: https://docs.opencv.org/4.10.0/d7/d53/tutorial_py_pose.html (besucht am 29.06.2024).
- [71] S. Garrido-Jurado u. a. „Automatic Generation and Detection of Highly Reliable Fiducial Markers under Occlusion“. In: *Pattern Recognition* 47.6 (Juni 2014), S. 2280–2292. ISSN: 00313203. DOI: 10.1016/j.patcog.2014.01.005. URL: <https://linkinghub.elsevier.com/retrieve/pii/S0031320314000235> (besucht am 27.07.2024).
- [72] F. Landis Markley u. a. „Averaging Quaternions“. In: *Journal of Guidance, Control, and Dynamics* 30.4 (Juli 2007), S. 1193–1197. ISSN: 0731-5090, 1533-3884. DOI: 10.2514/1.28949. URL: <https://arc.aiaa.org/doi/10.2514/1.28949> (besucht am 27.06.2024).
- [73] Dawson-Haggerty et al. *Trimesh*. Version 3.2.0. 8. Dez. 2019. URL: <https://trimesh.org/>.
- [74] Maximilian Denninger u. a. *BlenderProc*. 25. Okt. 2019. arXiv: 1911.01911 [cs]. URL: <http://arxiv.org/abs/1911.01911> (besucht am 21.12.2023). Vorveröffentlichung.
- [75] Maximilian Denninger u. a. „BlenderProc2: A Procedural Pipeline for Photorealistic Rendering“. In: *Journal of Open Source Software* 8.82 (20. Feb. 2023), S. 4901. ISSN: 2475-9066. DOI: 10.21105/joss.04901. URL: <https://joss.theoj.org/papers/10.21105/joss.04901> (besucht am 18.07.2024).
- [76] Hannes Sommer u. a. „Why and How to Avoid the Flipped Quaternion Multiplication“. In: *Aerospace* 5.3 (5. Juli 2018), S. 72. ISSN: 2226-4310. DOI: 10.3390/aerospace5030072. URL: <http://www.mdpi.com/2226-4310/5/3/72> (besucht am 29.04.2024).

A. Anhang

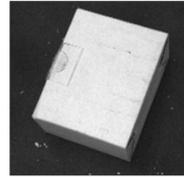
A.1. Objekte des ITODD Datensatzes [46]



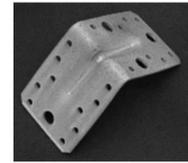
adapter_plate_square



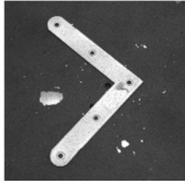
adapter_plate_triangular



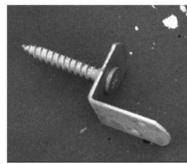
box



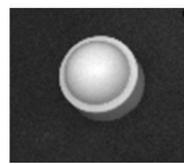
bracket_big



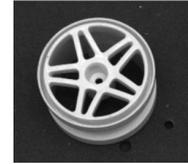
bracket_planar



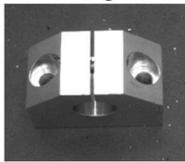
bracket_screw



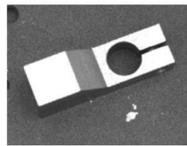
cap



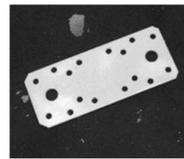
car_rim



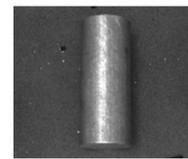
clamp_big



clamp_small



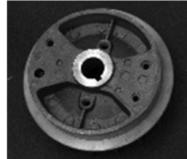
connector_plate



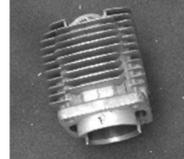
cylinder



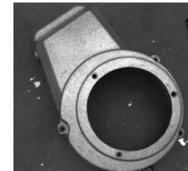
engine_part_bearing



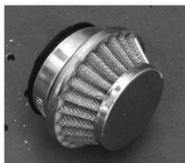
engine_part_cooler_round



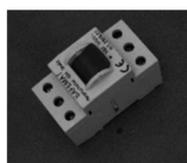
engine_part_cooler_square



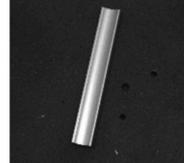
engine_part_cover



filter



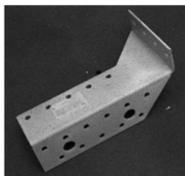
fuse



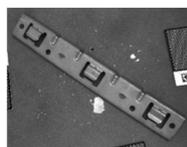
handle



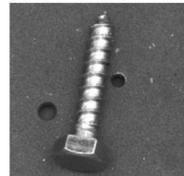
injection_pump



multi_bracket



punched_rail



screw



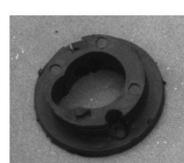
screw_black



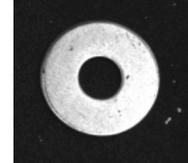
star



tee_connector



thread



washer

A.2. Ergebnisse der verwendeten Methoden in der BOP Challenge, Daten aus [36]

Date (UTC)	Method	Test image	AR _{Core}	AR _{LM-O}	AR _{T-LESS}	AR _{TUD-L}	AR _{IC-BIN}	AR _{ITODD}	AR _{HB}	AR _{YCB-V}	Time (s)	Object Type
2023-11-19	FoundationPose (NVI-DIA)	RGB-D	0.726	0.733	0.617	0.906	0.528	0.609	0.809	0.882		neuartig
2024-01-22	SAM6D	RGB-D	0.704	0.699	0.515	0.904	0.588	0.602	0.776	0.845	4.367	neuartig
2024-03-04	SAM6D-FastSAM	RGB-D	0.662	0.667	0.485	0.829	0.51	0.572	0.736	0.834	1.426	neuartig
2024-03-04	SAM6D-CNOSfastSAM	RGB-D	0.653	0.651	0.479	0.825	0.497	0.562	0.738	0.815	1.254	neuartig
2023-09-27	Megapose-CNOS_fast-SAM+Multihyp	RGB-D	0.628	0.626	0.487	0.851	0.467	0.468	0.73	0.764	141.965	neuartig
2023-09-27	Megapose-CNOS_fast-SAM+Multihyp	RGB-D	0.623	0.62	0.485	0.846	0.462	0.46	0.725	0.764	116.564	neuartig
2024-03-04	SAM6D-ZeroPose	RGB-D	0.622	0.635	0.43	0.802	0.518	0.484	0.691	0.792	5.482	neuartig
2023-09-28	SAM6D-BOP2023-CNOSmask	RGB-D	0.616	0.648	0.483	0.794	0.504	0.351	0.727	0.804	3.872	neuartig
2024-01-27	FoundPose+FeatRef+Megapose-5hyp	RGB	0.596	0.61	0.57	0.693	0.479	0.407	0.723	0.69	20.523	neuartig
2024-01-29	FoundPose+FeatRef+Megapose	RGB	0.55	0.556	0.511	0.633	0.433	0.357	0.697	0.661	6.385	neuartig
2023-09-26	Megapose-CNOS_fastSAM+MultiHyp-10	RGB	0.549	0.56	0.508	0.687	0.419	0.346	0.706	0.62	53.878	neuartig
2023-09-26	Megapose-CNOS_fast-SAM+MultiHyp	RGB	0.547	0.56	0.507	0.684	0.414	0.338	0.704	0.621	47.386	neuartig
2024-01-29	FoundPose+MegaPose	RGB	0.547	0.554	0.51	0.633	0.43	0.346	0.695	0.661	4.398	neuartig
2023-09-04	MegaPose-CNOS_fast-SAM	RGB	0.509	0.499	0.477	0.653	0.367	0.315	0.654	0.601	31.724	neuartig
2024-01-29	FoundPose+FeatRef	RGB	0.426	0.395	0.396	0.567	0.283	0.262	0.585	0.497	2.648	neuartig
2024-01-28	FoundPose-Coarse	RGB	0.373	0.397	0.338	0.469	0.239	0.204	0.508	0.452	1.69	neuartig
2023-11-02	MegaPose-CNOS_fast-SAM+CoarseBest	RGB	0.208	0.229	0.177	0.258	0.152	0.108	0.251	0.281	15.465	neuartig

A.3. Ergebnisse der Versuche

A.3.1. Versuch A1

object_name	occlusion	sensor	method	motion	AP_p	ap_x	ap_y	ap_z	ap_a	ap_b	ap_c	RP_i	RP_a	RP_b	RP_c
obj_000002	base	luxonis	FoundationPose	circ_far	12.52	-1.34	2.29	12.23	175.33	117.53	62.42	5.83	5.11	2.1	1.91
obj_000002	base	luxonis	FoundationPose	circ_close	18.94	-0.81	2.89	18.7	169.97	115.67	65.16	16.83	4.44	5.81	6.23
obj_000002	base	luxonis	FoundationPose	zoom	16.01	-1.56	0.94	15.91	1.65	1.7	1.72	9.44	2.76	3.11	2.06
obj_000002	base	luxonis	FoundationPose	all_motions	15.28	-1.25	2.1	15.09	126.67	85.57	46.74	13.61	229.89	154.19	82.88
obj_000002	base	realsense	FoundationPose	circ_far	4.09	0.53	3.76	-1.52	174.17	120.05	59.98	9.34	6.34	2.42	2.48
obj_000002	base	realsense	FoundationPose	circ_close	5.11	1.29	4.91	-0.57	169.52	120.23	60.72	3.65	3.14	2.49	2.3
obj_000002	base	realsense	FoundationPose	zoom	6.07	0.28	2.26	5.63	170.96	142.57	36.47	7.01	5.18	1.33	0.79
obj_000002	base	realsense	FoundationPose	all_motions	3.81	0.68	3.68	0.68	171.99	126.16	53.86	11.06	8.07	30.07	31.88
obj_000002	hand	luxonis	FoundationPose	circ_far	12.94	-0.9	0.93	12.87	121.94	90.21	47.31	8.46	167.92	113.23	54.48
obj_000002	hand	luxonis	FoundationPose	circ_close	323.74	23.23	-31.63	321.36	39.87	113.71	130.99	173.06	68.28	31.28	35.96
obj_000002	hand	luxonis	FoundationPose	zoom	34.14	6.58	-8.79	32.33	18.61	16.97	17.28	128.62	21.92	14.55	12.31
obj_000002	hand	luxonis	FoundationPose	all_motions	107.5	8.05	-11.04	106.63	70.37	77.08	63.16	336.7	182.89	137.89	140.93
obj_000002	hand	realsense	FoundationPose	circ_far	5.49	2.71	4.65	-1.05	128.91	94.35	48.14	14.79	179.66	120.37	59.63
obj_000002	hand	realsense	FoundationPose	circ_close	6.72	3.52	5.71	-0.37	114.62	86.92	45.97	6.13	176.49	122.22	67.33
obj_000002	hand	realsense	FoundationPose	zoom	12.61	10.71	-3.6	5.61	27.16	21.75	18.43	8.31	9.47	8.95	5.66
obj_000002	hand	realsense	FoundationPose	all_motions	5.85	5.07	2.77	0.91	97.75	72.92	39.62	16.99	199.4	139	65.89
obj_000002	approach	luxonis	FoundationPose	circ_far	30.07	-6.85	-0.07	29.28	48.74	25.32	53.39	70.54	124.22	60.11	141
obj_000002	approach	luxonis	FoundationPose	circ_close	21.84	8.97	12	15.89	142.46	130.06	57.98	46.44	56.78	55.81	26.93
obj_000002	approach	luxonis	FoundationPose	zoom	83.17	47.75	42.43	53.26	93.91	88.61	105.2	53.92	22.94	25.29	16.73
obj_000002	approach	luxonis	FoundationPose	all_motions	37.3	12.35	14.81	31.93	87.58	72.17	68.65	94.27	147.76	144.31	116.44
obj_000002	approach	realsense	FoundationPose	circ_far	6.13	1	5.88	-1.4	169.29	117.39	61.47	11.52	18.33	7.54	4.27
obj_000002	approach	realsense	FoundationPose	circ_close	7.69	2.27	7.27	-1.09	163.1	116.41	62.57	5.5	24.57	5.66	4.38
obj_000002	approach	realsense	FoundationPose	zoom	7.31	-0.18	4.58	5.69	6.31	5.94	4.38	5.83	13.32	10.64	6.53
obj_000002	approach	realsense	FoundationPose	all_motions	6.05	1.04	5.93	0.61	123.3	86.87	46.3	11.86	215.97	148.96	77.21
obj_000006	approach	luxonis	FoundationPose	circ_far	16.87	-10.17	6.78	11.62	8.73	10.79	12.27	25.94	14.84	23.75	23.15
obj_000006	approach	luxonis	FoundationPose	circ_close	39.41	-4.1	9.86	37.93	27.64	27.02	17.17	60.86	85.45	81.82	33.83
obj_000006	approach	luxonis	FoundationPose	zoom	12.3	-3.65	-1.37	11.67	7.4	8.21	12.38	16.74	5.61	39.62	35.42
obj_000006	approach	luxonis	FoundationPose	all_motions	20.98	-6.69	5.46	19.12	13.75	14.71	13.7	58.99	53.31	55.57	30.51
obj_000006	approach	realsense	FoundationPose	circ_far	9.24	-8.09	3.97	-2.03	21.41	15.79	24.61	23.35	44.17	23.35	37.36
obj_000006	approach	realsense	FoundationPose	circ_close	2.66	-2.4	0.57	0.98	20.62	22.28	28.19	20.78	58.92	49.17	66.06
obj_000006	approach	realsense	FoundationPose	zoom	12.41	-1.67	-11.92	2.99	28.36	121.36	125.27	35.69	22.18	19.77	20.49
obj_000006	approach	realsense	FoundationPose	all_motions	4.9	-4.74	-1.23	0.17	23.03	45.71	52.38	33.58	45.46	140.96	139.33
obj_000006	base	luxonis	FoundationPose	circ_far	14.76	-1.6	-0.09	14.67	4.37	6	6.06	8.12	6.86	13.36	14.84
obj_000006	base	luxonis	FoundationPose	circ_close	13.73	-0.74	2.36	13.51	4.27	6.1	6.5	6.84	7.29	13.19	13.86
obj_000006	base	luxonis	FoundationPose	zoom	19.34	-2.71	-2.49	18.98	4.6	32.91	32.94	10.6	5.08	9.62	9.54
obj_000006	base	luxonis	FoundationPose	all_motions	15.6	-1.65	-0.04	15.51	4.4	13.33	13.48	10.95	6.52	38.02	38.13
obj_000006	base	realsense	FoundationPose	circ_far	3.24	-0.5	-1.74	-2.68	4.3	24.45	24.41	20.15	9.25	66.3	66.65
obj_000006	base	realsense	FoundationPose	circ_close	2.46	0.07	1.04	2.23	4.21	7.96	7.64	6.79	9.34	24.88	27.81
obj_000006	base	realsense	FoundationPose	zoom	12.22	-1.7	-4.16	11.37	7.75	53.32	53.78	10.47	3.39	19.32	19.12
obj_000006	base	realsense	FoundationPose	all_motions	3.05	-0.67	-1.62	2.5	5.21	27.68	27.7	23	9.31	69.79	70.88
obj_000006	hand	luxonis	FoundationPose	circ_far	14.57	-0.14	5.67	13.43	71.94	57.52	40.52	32.75	45.25	41.43	19.56
obj_000006	hand	luxonis	FoundationPose	circ_close	28.91	18.54	-10.02	19.79	91.87	111.63	73.52	53.78	89.62	39.21	68.31
obj_000006	hand	luxonis	FoundationPose	zoom	29.65	-4.81	-15.02	25.11	27.49	152.63	168.38	59.6	26.97	13.3	9.35
obj_000006	hand	luxonis	FoundationPose	all_motions	19.41	3.88	-4.52	18.47	65.3	99.26	85.32	52.66	94.42	125.3	164.81
obj_000006	hand	realsense	FoundationPose	circ_far	8.43	-8.32	1	-0.98	11.37	10.84	12.76	14.68	20.82	22.01	22.24
obj_000006	hand	realsense	FoundationPose	circ_close	20.48	18.65	-8.4	-0.88	95.79	117.14	71.31	27.73	92.63	35.61	83.41
obj_000006	hand	realsense	FoundationPose	zoom	21.61	-19.35	3.53	8.96	89.53	102.85	136.71	10.73	10.5	7.25	20.46
obj_000006	hand	realsense	FoundationPose	all_motions	4.12	-3.61	-0.99	1.73	56.44	65.87	62.8	44.22	132.32	151.53	161.01
obj_000014	base	luxonis	FoundationPose	circ_far	5.39	-1.43	1.59	4.95	1.84	2.06	1.64	5.12	2.55	3.53	2.88
obj_000014	base	luxonis	FoundationPose	circ_close	221.36	63.75	25.61	210.43	41.88	79.43	70.07	141.55	77.49	34.47	49.5
obj_000014	base	luxonis	FoundationPose	zoom	11.32	-0.15	-1.41	11.23	177.85	141.42	38.58	23.7	3.31	6.99	7.13
obj_000014	base	luxonis	FoundationPose	all_motions	68.35	17.61	7.67	65.59	61.08	62.06	31.29	255.97	224.22	176.25	90.95
obj_000014	base	realsense	FoundationPose	circ_far	7.06	0.12	2.3	-6.67	1.18	1.26	1.59	6.95	1.94	2.32	2.64
obj_000014	base	realsense	FoundationPose	circ_close	6.53	0.57	3.19	-5.67	169.94	119.93	61.09	4.03	3.61	3.41	2.94
obj_000014	base	realsense	FoundationPose	zoom	1.11	-0.74	0.73	0.39	178.55	143.2	36.83	5.35	1.58	2.12	2.06
obj_000014	base	realsense	FoundationPose	all_motions	4.96	0.01	2.12	-4.48	96.4	72.89	27.76	9.29	259.38	196.72	76.24

obj_000014	hand	luxonis	FoundationPose	circ_far	166.06	-106.33	45.35	119.21	106.98	126.5	102.75	143.21	54.43	117.03	97.47
obj_000014	hand	luxonis	FoundationPose	circ_close	495.84	9.81	29.66	494.85	138.33	138.99	27.7	261.63	59.8	53.39	30.85
obj_000014	hand	luxonis	FoundationPose	zoom	200.74	-14.86	-43.58	195.38	4.31	51.31	51.37	327.96	9.69	42.94	42.83
obj_000014	hand	luxonis	FoundationPose	all_motions	252.82	-48.2	16.72	247.62	88.11	109.68	67.29	409.13	166.12	137.97	121.05
obj_000014	hand	realsense	FoundationPose	circ_far	30.07	-6.85	-0.07	29.28	48.74	25.32	53.39	70.54	124.22	60.11	141
obj_000014	hand	realsense	FoundationPose	circ_close	21.84	8.97	12	15.89	142.46	130.06	57.98	46.44	56.78	55.81	26.93
obj_000014	hand	realsense	FoundationPose	zoom	83.17	47.75	42.43	53.26	93.91	88.61	105.2	53.92	22.94	25.29	16.73
obj_000014	hand	realsense	FoundationPose	all_motions	37.3	12.35	14.81	31.93	87.58	72.17	68.65	94.27	147.76	144.31	116.44
obj_000014	approach	luxonis	FoundationPose	circ_far	4.94	-1.18	0.59	4.76	175.73	119.81	60.34	6.27	6.84	4.88	4.86
obj_000014	approach	luxonis	FoundationPose	circ_close	488.48	-109.69	51.49	473.21	101.27	63.76	122.75	501.06	70.92	64.3	24.25
obj_000014	approach	luxonis	FoundationPose	zoom	44.56	-2.48	1.53	44.46	174.94	141.88	40.04	244.8	27.68	23.39	10.61
obj_000014	approach	luxonis	FoundationPose	all_motions	154.19	-32.66	15.45	149.89	154.16	109.72	72.73	616.12	108.82	98.82	99.8
obj_000014	approach	realsense	FoundationPose	circ_far	7.17	0.32	2.61	-6.67	1.74	1.64	2.26	6.63	3.13	2.87	3.53
obj_000014	approach	realsense	FoundationPose	circ_close	6.02	0.44	2.53	-5.45	170.07	120.43	60.57	4.72	3.83	5.09	4.88
obj_000014	approach	realsense	FoundationPose	zoom	1.34	-0.46	1.25	0.14	178.4	142.54	37.5	6.2	1.65	3.06	3
obj_000014	approach	realsense	FoundationPose	all_motions	5.03	0.15	2.23	-4.51	97.33	73.4	28.48	8.77	258.12	195.27	75.4
obj_000021	approach	luxonis	FoundationPose	circ_far	3.95	0.56	1.5	3.61	2.38	2.27	2.34	15.29	3.63	2.39	3.2
obj_000021	approach	luxonis	FoundationPose	circ_close	11.22	-0.67	4.12	10.42	3.91	7.06	6.95	21.11	5.58	17.98	17.63
obj_000021	approach	luxonis	FoundationPose	zoom	7.78	-1.05	-0.14	7.71	1.03	1.74	1.75	14.2	1.91	3.1	3.26
obj_000021	approach	luxonis	FoundationPose	all_motions	6.92	-0.23	1.81	6.68	2.45	3.5	3.5	18.78	5.1	11.94	11.77
obj_000021	approach	realsense	FoundationPose	circ_far	7.84	-0.45	2.01	-7.57	3.21	3.22	4	12.03	5.19	6.77	6.46
obj_000021	approach	realsense	FoundationPose	circ_close	12.59	-2.99	4.9	-11.2	6.07	10.12	10.63	20.99	17.77	39.53	40.48
obj_000021	approach	realsense	FoundationPose	zoom	2.4	-1.23	0.11	-2.06	1.93	2.63	2.86	9.26	1.58	1.56	2.33
obj_000021	approach	realsense	FoundationPose	all_motions	7.62	-1.38	2.32	-7.12	3.68	5.02	5.58	19.11	11.13	23.48	23.9
obj_000021	hand	luxonis	FoundationPose	circ_far	7.38	-1.15	3	6.65	2.64	3.37	3	17.08	4.04	4.58	5.71
obj_000021	hand	luxonis	FoundationPose	circ_close	8.69	-0.17	5.9	6.38	3.56	4.5	4.15	12.95	8	5.98	5.8
obj_000021	hand	luxonis	FoundationPose	zoom	4.92	-2.64	1.97	3.65	2.17	1.62	1.82	19.94	2.71	1.52	3.54
obj_000021	hand	luxonis	FoundationPose	all_motions	6.88	-1.28	3.55	5.76	2.78	3.22	3.01	16.65	5.45	5.52	5.83
obj_000021	hand	realsense	FoundationPose	circ_far	9.06	-0.6	4.09	-8.06	7.11	10.19	11.46	13.91	20.4	27.25	33.99
obj_000021	hand	realsense	FoundationPose	circ_close	5.93	0.94	3.58	-4.63	2.76	3.84	4.54	12.33	4.36	9.12	9.3
obj_000021	hand	realsense	FoundationPose	zoom	33.23	6.21	11.46	30.56	95.82	153.55	89.93	7.99	4.95	9.32	8.04
obj_000021	hand	realsense	FoundationPose	all_motions	7.05	1.69	5.94	3.4	29.93	47.27	30.77	41.27	122.01	196.43	111.55
obj_000021	base	luxonis	FoundationPose	circ_far	11.94	-0.46	2.06	11.75	1.51	1.91	1.81	10.38	1.57	2.44	2.67
obj_000021	base	luxonis	FoundationPose	circ_close	11.23	0.51	4.11	10.44	2.04	2.07	1.87	5.91	1.86	1.78	2.73
obj_000021	base	luxonis	FoundationPose	zoom	13.69	-0.95	1.22	13.61	0.4	0.56	0.66	10.25	0.62	1.17	1
obj_000021	base	luxonis	FoundationPose	all_motions	12.13	-0.31	2.42	11.88	1.36	1.59	1.52	10.15	2.39	2.74	2.82
obj_000021	base	realsense	FoundationPose	circ_far	2.43	0.66	1.79	-1.51	1.25	0.85	1.49	5.07	1.9	2.29	2.56
obj_000021	base	realsense	FoundationPose	circ_close	3.79	1.73	3.09	-1.37	0.92	0.72	1.19	4.69	1.78	1.3	1.67
obj_000021	base	realsense	FoundationPose	zoom	3.83	-1.22	-0.19	3.62	1.08	0.76	1.21	7.36	1.7	1.23	1.58
obj_000021	base	realsense	FoundationPose	all_motions	1.69	0.46	1.62	-0.08	1.11	0.79	1.33	8.39	1.85	1.79	2.13
obj_000002	base	luxonis	MegaPose	circ_far	6.76	-1.31	2.26	6.24	83.9	56.75	30.1	6.47	262.76	175.58	91.81
obj_000002	base	luxonis	MegaPose	circ_close	8.95	-0.44	4.15	7.92	109.49	77.06	40.91	5.32	240.8	167.45	88.38
obj_000002	base	luxonis	MegaPose	zoom	102.51	-1.88	8.18	102.17	91.73	109.49	134.96	125.41	244.77	169.6	126.77
obj_000002	base	luxonis	MegaPose	all_motions	33.07	-1.22	4.41	32.75	93.36	76.88	61.65	125.85	251.93	182.36	168.78
obj_000002	base	realsense	MegaPose	circ_far	3.16	-0.02	2.31	2.16	61.14	42.5	21.86	6.93	246.63	169.69	83.87
obj_000002	base	realsense	MegaPose	circ_close	4.79	0.95	3.76	2.81	105.88	75.39	38.76	4.17	244.73	172.1	86.93
obj_000002	base	realsense	MegaPose	zoom	79.76	2.04	9.29	79.19	111.89	114.72	114.28	127.55	248.38	170.29	172.68
obj_000002	base	realsense	MegaPose	all_motions	23.55	0.81	4.6	23.08	87.54	71.3	51.55	128.5	254.93	191.41	163.33
obj_000002	hand	luxonis	MegaPose	circ_far	20.56	-2.35	-9.33	18.17	65.38	53.24	28.87	60.48	226.55	167.06	93.19
obj_000002	hand	luxonis	MegaPose	circ_close	20.98	-1.87	-6.63	19.82	79.37	71.14	42.62	49.41	236.34	176.53	125.19
obj_000002	hand	luxonis	MegaPose	zoom	157.4	2.2	-8.99	157.13	11.42	167.58	169.35	162.74	83.71	63.58	26.76
obj_000002	hand	luxonis	MegaPose	all_motions	56.98	-0.98	-8.46	56.34	54.75	89.4	70.93	189.82	216.05	207.84	203.5
obj_000002	hand	realsense	MegaPose	circ_far	22.41	0.44	-13.45	17.92	52.54	57.41	37.81	68.71	214.61	162.56	118.61
obj_000002	hand	realsense	MegaPose	circ_close	15.98	1.96	-9.79	12.47	79.61	67.72	33.1	23.97	234.29	159.72	57.14
obj_000002	hand	realsense	MegaPose	zoom	139.83	10.91	3.47	139.36	9.86	156.44	156.05	178.14	8.47	112.39	115.02
obj_000002	hand	realsense	MegaPose	all_motions	49.38	3.66	-7.9	48.6	49.03	86.69	67.85	186.2	205.15	195.69	190.05
obj_000002	approach	luxonis	MegaPose	circ_far	9.14	-2.01	0.21	8.91	109.3	78.24	40.44	12.95	236.09	161.22	80.6
obj_000002	approach	luxonis	MegaPose	circ_close	10.61	-1.05	2.63	10.23	101.72	74.05	37.68	10.28	233.6	165.04	83.15
obj_000002	approach	luxonis	MegaPose	zoom	78.11	0.73	10.49	77.4	107.57	110.47	130.73	128.28	248.61	164.47	128.96
obj_000002	approach	luxonis	MegaPose	all_motions	28.13	-0.99	3.7	27.87	106.65	85.78	64.15	111.95	237.16	168.24	155.53
obj_000002	approach	realsense	MegaPose	circ_far	8.44	-0.66	-1.07	8.35	115.59	83.21	46.13	53.72	232.66	159.38	101.08
obj_000002	approach	realsense	MegaPose	circ_close	4.62	0.46	2.57	3.8	99.8	72.18	35.99	4.99	241.84	171.58	80.72
obj_000002	approach	realsense	MegaPose	zoom	81.55	3.42	11.86	80.61	107.83	115.9	126.31	130.42	240.85	163.84	145.13

obj_000002	approach	realsense	MegaPose	all_motions	26.89	0.77	3.49	26.65	108.96	88.92	64.97	119.52	236.51	170.81	157.36
obj_000006	approach	luxonis	MegaPose	circ_far	14.48	1.06	-5.47	-13.36	57.61	75.17	77.38	198.07	139.61	101.12	108.06
obj_000006	approach	luxonis	MegaPose	circ_close	6.35	1.96	-5.03	-3.35	45.39	68.76	67.02	149.98	125.65	118.08	130.22
obj_000006	approach	luxonis	MegaPose	zoom	65.49	-1.93	-3.96	-65.34	32.95	49.31	58.63	114	98.65	108.71	111.22
obj_000006	approach	luxonis	MegaPose	all_motions	25.01	0.51	-4.94	-24.51	47.49	66.38	69.38	187.92	128.55	112.08	117.08
obj_000006	approach	realsense	MegaPose	circ_far	9.84	1.14	-8.58	4.69	62.25	74.22	87.18	207.83	144.66	95.79	102.5
obj_000006	approach	realsense	MegaPose	circ_close	9.63	0.44	-5.01	-8.21	49.9	66.97	72.81	133.14	112.65	106.77	89.59
obj_000006	approach	realsense	MegaPose	zoom	60.58	-4.11	-4.84	-60.24	23.88	51.51	57.76	109.42	55.32	86.6	89.08
obj_000006	approach	realsense	MegaPose	all_motions	17.56	-0.46	-6.56	-16.29	48.49	66.09	75.21	187.49	125.89	100.04	101.47
obj_000006	base	luxonis	MegaPose	circ_far	10.49	-2.27	-1.52	10.12	30.86	50.41	65.32	64.96	114.11	94.08	139.83
obj_000006	base	luxonis	MegaPose	circ_close	10.02	-1.67	0.54	9.87	27.38	41.9	55.23	57.52	73.85	89.13	122.56
obj_000006	base	luxonis	MegaPose	zoom	11.05	-0.77	-3.76	10.36	12.08	18.61	22.63	62.59	66.1	48.15	62.27
obj_000006	base	luxonis	MegaPose	all_motions	10.37	-1.69	-1.54	10.11	24.76	39.34	50.84	61.91	94.62	91.04	129.07
obj_000006	base	realsense	MegaPose	circ_far	25.51	-0.86	-4.22	25.14	18.78	43.81	50.78	105.09	70.23	75.17	72.54
obj_000006	base	realsense	MegaPose	circ_close	11.06	-0.92	-0.33	11.02	16.89	28.58	34.65	58.48	35.56	56.44	51.76
obj_000006	base	realsense	MegaPose	zoom	17.08	-2.64	-3.11	16.59	10.85	20.46	21.1	82.94	62.63	95.98	85.76
obj_000006	base	realsense	MegaPose	all_motions	19.14	-1.36	-2.83	18.88	16.1	33.22	38.23	94.76	60.69	82.14	80.16
obj_000006	hand	luxonis	MegaPose	circ_far	33	2.18	-0.06	-32.93	67.04	83.58	82.22	178.94	104.84	113.55	92.99
obj_000006	hand	luxonis	MegaPose	circ_close	2.07	1.48	-1.44	0.17	75.79	71.48	78.53	117.9	149.11	101.72	87.27
obj_000006	hand	luxonis	MegaPose	zoom	22.25	-1.5	-8.75	-20.4	78.18	59.87	93.94	108.45	177.01	167.76	159.13
obj_000006	hand	luxonis	MegaPose	all_motions	20.27	0.96	-2.86	-20.04	72.62	73.57	84.42	152.34	140.38	130.39	114.39
obj_000006	hand	realsense	MegaPose	circ_far	32.57	-1.42	-4.21	-32.27	62.14	82.16	83.7	168.72	100.05	136.86	91.69
obj_000006	hand	realsense	MegaPose	circ_close	34.5	-3.08	-6.96	-33.65	77.39	97.07	83.39	131.51	101.75	126.45	68.96
obj_000006	hand	realsense	MegaPose	zoom	58.04	-4.5	-9.12	-57.14	45.62	101.91	79.44	173.51	105.72	200.63	150.58
obj_000006	hand	realsense	MegaPose	all_motions	39.95	-2.72	-6.31	-39.36	62.03	91.72	82.47	168.71	107.35	155.07	105.11
obj_000014	base	luxonis	MegaPose	circ_far	5.48	-1.26	2.37	-4.78	41.15	27.73	14.94	8.22	222.31	149.81	76.39
obj_000014	base	luxonis	MegaPose	circ_close	3.21	-0.51	2.98	1.07	6.13	4.63	3.02	4.92	84.24	57.57	30.17
obj_000014	base	luxonis	MegaPose	zoom	10.54	-1.32	0.37	-10.45	41.76	32.85	9.79	15.12	227.29	179.96	47.67
obj_000014	base	luxonis	MegaPose	all_motions	5.17	-1.06	2	-4.64	31.27	22.49	10.12	14.9	199.11	143.38	60.31
obj_000014	base	realsense	MegaPose	circ_far	8.29	-0.28	2.09	-8.02	31.34	21.45	11.52	5.65	198.99	136.16	67.25
obj_000014	base	realsense	MegaPose	circ_close	5.35	0.85	2.6	-4.6	15.43	10.87	6.56	3.84	141.21	100.33	49.9
obj_000014	base	realsense	MegaPose	zoom	8.38	-1.4	0.04	-8.26	102.2	81.88	22	8.31	265.48	213.55	54.21
obj_000014	base	realsense	MegaPose	all_motions	7.33	-0.27	1.68	-7.13	46.13	34.89	12.98	7.85	229.83	175.36	61.55
obj_000014	hand	luxonis	MegaPose	circ_far	1.95	0.65	-0.24	-1.82	78.54	56.94	28.82	85.01	251.31	181.11	86.77
obj_000014	hand	luxonis	MegaPose	circ_close	11.24	-0.59	-9.25	6.37	36.5	30.68	22.86	57.62	183.59	130.14	70.31
obj_000014	hand	luxonis	MegaPose	zoom	122.36	-7.05	-16.36	121.06	110.11	93.91	43.52	425.23	202.27	171.27	102.91
obj_000014	hand	luxonis	MegaPose	all_motions	34.67	-1.79	-7.19	33.87	75.05	59.44	31.1	339.85	234.35	178.91	89.86
obj_000014	hand	realsense	MegaPose	circ_far	6.49	0.79	-4.64	4.47	50.8	38.08	21.84	91.55	222.54	155.8	83.27
obj_000014	hand	realsense	MegaPose	circ_close	4.49	1.58	-2.72	3.2	84.04	63.07	30.68	20.74	254.15	180.67	79.08
obj_000014	hand	realsense	MegaPose	zoom	168.21	5.69	-14.71	167.47	75.01	68.8	36.42	357.01	210.35	176.24	108.81
obj_000014	hand	realsense	MegaPose	all_motions	48.53	2.33	-6.8	47.99	66.78	53.46	28.28	340.52	231.39	172.61	90.96
obj_000014	approach	luxonis	MegaPose	circ_far	5.02	-1.16	2.1	-4.41	10.7	7.61	4.81	7.98	119.57	78.77	40.9
obj_000014	approach	luxonis	MegaPose	circ_close	3.15	-0.64	2.81	1.26	24.74	17.8	9.68	4.79	176.19	121.72	62.6
obj_000014	approach	luxonis	MegaPose	zoom	11.51	-1.63	0.08	-11.4	11.21	9.01	3.38	14.77	125.82	99.38	26.23
obj_000014	approach	luxonis	MegaPose	all_motions	5.13	-1.14	1.75	-4.68	14.86	10.91	5.82	15.07	139.81	98.39	45.54
obj_000014	approach	realsense	MegaPose	circ_far	8.65	-0.08	2.01	-8.42	29.13	20.12	11.04	6.14	192.41	130.9	64.41
obj_000014	approach	realsense	MegaPose	circ_close	5.54	0.72	2.39	-4.94	24.11	17.25	9.74	4.27	174.68	123.22	62.11
obj_000014	approach	realsense	MegaPose	zoom	7.77	-1.78	-0.26	-7.56	94.97	76.38	20.03	8.44	268.48	216.36	53.73
obj_000014	approach	realsense	MegaPose	all_motions	7.35	-0.3	1.52	-7.18	45.17	34.24	13.05	8.02	227.78	172.81	61.93
obj_000021	approach	luxonis	MegaPose	circ_far	17.98	4.28	-1.1	-17.43	68.29	61.28	83.28	238.28	162.89	145.23	166.08
obj_000021	approach	luxonis	MegaPose	circ_close	12.85	-2.52	-7.51	10.11	65.88	59.79	81.7	277.78	154.46	135.26	166.66
obj_000021	approach	luxonis	MegaPose	zoom	90.2	4.35	-5.92	89.9	82.79	69.95	81.57	312.76	195.39	131.32	123.8
obj_000021	approach	luxonis	MegaPose	all_motions	20.18	2.35	-4.25	19.59	71.53	63.2	82.36	316.04	169.99	138.24	154.78
obj_000021	approach	realsense	MegaPose	circ_far	3.19	1.39	-0.09	2.87	43.16	47.11	50.7	193.74	157.5	139.8	143.22
obj_000021	approach	realsense	MegaPose	circ_close	22.46	4.93	-1.66	-21.84	62.28	59.68	78.32	266.63	167.42	129.41	160.62
obj_000021	approach	realsense	MegaPose	zoom	50.31	-1.75	-2.44	50.22	74.44	57.53	47.5	193.19	228.3	186.59	159.59
obj_000021	approach	realsense	MegaPose	all_motions	8.8	1.56	-1.17	8.58	57.02	53.49	57.7	235.82	184.69	150.96	156.61
obj_000021	hand	luxonis	MegaPose	circ_far	13.5	5.28	2.55	-12.16	69.51	74.49	75.38	376.2	158.51	156.46	155.09
obj_000021	hand	luxonis	MegaPose	circ_close	11.99	1.54	-3.32	-11.41	73.28	60.07	75.89	282.61	156.77	119.66	152.63
obj_000021	hand	luxonis	MegaPose	zoom	50.66	-10.02	-0.99	49.65	37.07	104.96	107.98	293.82	127.3	198.14	174.9
obj_000021	hand	luxonis	MegaPose	all_motions	4.83	0.05	-0.1	4.82	61.79	78.62	84.37	338.97	155.87	166.84	164.57
obj_000021	hand	realsense	MegaPose	circ_far	2.28	2.1	-0.32	-0.83	43.04	56.43	64.08	326.22	131.57	144.56	167.58
obj_000021	hand	realsense	MegaPose	circ_close	21.65	0.41	-2.45	-21.5	67.48	59.03	70.59	251.75	172.44	137.39	147.64

obj_000021	hand	realsense	MegaPose	zoom	180.73	16.1	1.53	180.01	54.95	95.85	76.02	177.85	116.68	78.29	81.41
obj_000021	hand	realsense	MegaPose	all_motions	42.65	5.41	-0.43	42.31	53.28	67.87	69.19	308.27	143.19	136.87	142.83
obj_000021	base	luxonis	MegaPose	circ_far	6.66	0.63	3.74	5.47	8.33	9.06	13.25	94.87	47.1	62.32	73.52
obj_000021	base	luxonis	MegaPose	circ_close	9.17	0.96	4.57	7.89	6.9	9.27	12.17	94.7	47.3	69.46	81.4
obj_000021	base	luxonis	MegaPose	zoom	17.71	0.66	-3.57	-17.34	3.17	8.38	9.4	95.15	25.27	13.37	24.2
obj_000021	base	luxonis	MegaPose	all_motions	2.13	0.73	2	-0.02	6.52	8.94	11.9	100.15	42.55	55.72	66.39
obj_000021	base	realsense	MegaPose	circ_far	3.33	1.65	2.88	0.28	5.8	6.08	8.78	45.54	33.47	27.93	39.13
obj_000021	base	realsense	MegaPose	circ_close	4.79	1.97	4.27	-0.94	9.5	6.73	12.01	38.67	57.46	25.81	56.99
obj_000021	base	realsense	MegaPose	zoom	8.58	0.59	-2.88	-8.07	5.86	6.17	9.13	27.75	46.08	11.03	42.74
obj_000021	base	realsense	MegaPose	all_motions	3.24	1.46	1.71	-2.33	6.88	6.29	9.8	41.66	44.81	23.7	45.68
obj_000002	base	luxonis	SAM6D	circ_far	15.35	-1.89	0.32	15.23	41.93	30.36	18.33	34.4	195.57	133.48	82.26
obj_000002	base	luxonis	SAM6D	circ_close	288.85	-23.9	-65.88	280.22	75.11	87.83	57.96	1209.68	156.93	135.44	132.52
obj_000002	base	luxonis	SAM6D	zoom	57.9	15.9	15.61	53.44	69.19	72.07	38.51	479.76	187.56	160.26	113.21
obj_000002	base	luxonis	SAM6D	all_motions	102.69	-3.37	-14.52	101.6	58.85	58.16	35.17	836.82	187.36	160.12	117.55
obj_000002	base	realsense	SAM6D	circ_far	4.54	0.96	4.37	-0.77	37.76	27.01	14.3	9.28	197.28	134.67	68.63
obj_000002	base	realsense	SAM6D	circ_close	6.82	1.98	6.53	0.11	61.03	44.59	23.96	5.8	228.03	161.34	84.61
obj_000002	base	realsense	SAM6D	zoom	4.63	-1.21	3.15	-3.17	108.75	89.64	32.53	104.44	221.15	180.8	98.94
obj_000002	base	realsense	SAM6D	all_motions	4.85	0.67	4.66	-1.17	63.49	48.88	21.96	56.01	228.68	173.01	84.73
obj_000002	hand	luxonis	SAM6D	circ_far	12.43	3.85	8.34	8.37	68.65	54.22	31.57	198.45	171.32	123.93	82.27
obj_000002	hand	luxonis	SAM6D	circ_close	473.1	24.38	-114.67	458.35	95.22	99.24	49.83	1692.9	160.71	141.57	115.66
obj_000002	hand	luxonis	SAM6D	zoom	160.71	71.91	61.98	129.67	80.82	74.44	46.23	1031.38	147.53	141.24	134.6
obj_000002	hand	luxonis	SAM6D	all_motions	172.01	26.83	-13.96	169.33	79.41	72.35	40.53	1260.36	164.82	144.38	109.28
obj_000002	hand	realsense	SAM6D	circ_far	28.15	18.6	20.86	-3.31	77.17	63.46	39.33	311.07	175.86	125.52	83.31
obj_000002	hand	realsense	SAM6D	circ_close	3.73	1.93	3.19	-0.01	78.29	62.07	34.15	41.45	176.91	129.05	66.03
obj_000002	hand	realsense	SAM6D	zoom	15.98	12.02	10.42	-1.58	90.29	85.18	33.28	280.12	175.09	150.73	93.14
obj_000002	hand	realsense	SAM6D	all_motions	17.83	12.03	13.02	-1.9	80.83	68.57	36.28	257.77	175.35	135.44	81.16
obj_000002	approach	luxonis	SAM6D	circ_far	12.92	-1.63	-1.26	12.75	51.31	42.22	26.77	83.66	168.9	129.79	88.28
obj_000002	approach	luxonis	SAM6D	circ_close	496.97	3.47	-112.61	484.03	81.8	90.62	41.99	1647.3	150.52	125.58	58.88
obj_000002	approach	luxonis	SAM6D	zoom	151.56	46.02	24.83	142.25	90.54	83.01	45.97	1253.92	180.81	145.64	123.11
obj_000002	approach	luxonis	SAM6D	all_motions	182.94	12.83	-25.44	180.71	70.61	66.99	36.3	1292.31	174.31	148.29	95.55
obj_000002	approach	realsense	SAM6D	circ_far	6.52	1.51	6.23	-1.16	57.78	44.47	25.72	11.22	186.07	129.2	71.43
obj_000002	approach	realsense	SAM6D	circ_close	6.48	2.43	6.01	0.06	58.57	45.34	25.19	5.89	192.42	139.89	73.17
obj_000002	approach	realsense	SAM6D	zoom	6.3	-0.06	3.19	5.44	101.63	86.11	27.44	35.27	214.09	175.19	46.7
obj_000002	approach	realsense	SAM6D	all_motions	5.61	1.36	5.36	0.95	69.66	55.78	26.02	22.23	202.55	154.6	65.88
obj_000006	approach	luxonis	SAM6D	circ_far	45.45	16.19	21.71	36.49	95.62	90.16	85.14	750.03	121.38	121.06	105.44
obj_000006	approach	luxonis	SAM6D	circ_close	71.69	64.93	-6.39	29.71	85.87	81.2	82.58	344.85	144.32	139.56	124.78
obj_000006	approach	luxonis	SAM6D	zoom	82.95	81.72	-10.03	10.12	69.81	92.11	77.6	375.42	137.22	83.93	66.19
obj_000006	approach	luxonis	SAM6D	all_motions	55.89	48.42	4.71	27.52	85.87	87.89	82.38	576.78	135.51	118.88	103.24
obj_000006	approach	realsense	SAM6D	circ_far	35.18	-20.93	15.5	23.65	91.71	96.48	94.92	967.89	129.42	123.49	90.86
obj_000006	approach	realsense	SAM6D	circ_close	17.27	2.81	15.62	6.81	85.98	80.18	77.81	343.05	121.76	144.2	85.91
obj_000006	approach	realsense	SAM6D	zoom	78.36	69.52	6.48	35.57	90.03	92.79	86.24	712.57	162.06	75.99	75.92
obj_000006	approach	realsense	SAM6D	all_motions	26.78	9	13.28	21.44	89.52	90.53	87.47	776.58	135.13	121.72	87.93
obj_000006	base	luxonis	SAM6D	circ_far	71.7	35.92	10.46	61.17	101.5	87.37	88.36	1144.14	148.83	129.22	110.73
obj_000006	base	luxonis	SAM6D	circ_close	47.66	41.3	4.66	23.31	81.81	74.54	86.37	263.9	161.02	142.72	142.91
obj_000006	base	luxonis	SAM6D	zoom	78.08	44.79	-21.14	60.36	97.22	88.22	86.05	818.41	188.8	78.78	76.14
obj_000006	base	luxonis	SAM6D	all_motions	63.4	39.87	0.51	49.29	94.32	83.63	87.15	882.66	164.26	123.36	113.6
obj_000006	base	realsense	SAM6D	circ_far	61.54	45.42	40.56	-8.87	96.27	88.05	87.98	357.13	148.33	160.95	117.42
obj_000006	base	realsense	SAM6D	circ_close	33.66	30.03	14.77	3.58	83	83.68	82.57	298.33	160.63	156.03	136.52
obj_000006	base	realsense	SAM6D	zoom	51.79	48.61	10.87	14.19	90.48	92.72	91.77	340.91	234.23	86.74	74.65
obj_000006	base	realsense	SAM6D	all_motions	49.04	41.99	25.31	0.86	91	88.1	87.5	345.78	178.05	142.46	113.24
obj_000006	hand	luxonis	SAM6D	circ_far	102.23	69.69	-1.89	74.77	83.54	89.54	78.89	1016.59	104.55	100.19	75.4
obj_000006	hand	luxonis	SAM6D	circ_close	91.95	90.22	-4.93	17.07	89.24	92.81	76.19	335.21	103.35	97.31	87.3
obj_000006	hand	luxonis	SAM6D	zoom	88.79	86.14	-8.74	19.69	89.84	83.12	89.24	342.33	144.43	98.64	61.18
obj_000006	hand	luxonis	SAM6D	all_motions	90.87	80.78	-4.66	41.34	87.06	89	80.62	733.78	114.83	98.56	77.34
obj_000006	hand	realsense	SAM6D	circ_far	83.7	11.55	75.51	-34.22	87.54	99.42	79.47	310.46	125.07	100.18	81.46
obj_000006	hand	realsense	SAM6D	circ_close	58.73	54.77	-15.9	14.03	86.25	103.07	81.88	368.13	112.85	113.74	79.11
obj_000006	hand	realsense	SAM6D	zoom	55.23	50.06	11.14	20.5	93.94	90.3	85.39	568.18	148.88	93.59	92.19
obj_000006	hand	realsense	SAM6D	all_motions	46.67	35.1	30.38	-4.7	88.91	98.01	81.82	433.16	127.94	102.97	83.47
obj_000014	base	luxonis	SAM6D	circ_far	7.18	-1.64	1.22	6.89	36.51	25.19	13.64	7.1	208.18	139.84	71.6
obj_000014	base	luxonis	SAM6D	circ_close	733.69	-39.97	-102.28	725.43	91.34	94.93	45.43	2508.35	159.51	146.08	93.54
obj_000014	base	luxonis	SAM6D	zoom	103.16	37.29	9.72	95.7	94.58	79.67	75.28	1122.95	150.35	134.8	160.67
obj_000014	base	luxonis	SAM6D	all_motions	240.33	-2.08	-26.37	238.87	68.24	60.24	39.68	1777.81	197.86	168.37	132.41
obj_000014	base	realsense	SAM6D	circ_far	6.11	0.92	2.72	-5.39	10.53	6.94	4.53	7.47	117.63	79.14	40.05

obj_000014	base	realsense	SAM6D	circ_close	6.19	1.82	3.76	-4.56	67.06	46.97	25.17	4.34	248.77	175.04	89.38
obj_000014	base	realsense	SAM6D	zoom	12.94	-8.67	9.55	0.99	61.24	47.67	26.07	233.12	222.36	179.71	116.57
obj_000014	base	realsense	SAM6D	all_motions	6.12	-1.43	4.87	-3.43	40.07	29.16	16.14	127.46	206.69	152.99	86.64
obj_000014	hand	luxonis	SAM6D	circ_far	204.99	96.22	-20.31	179.86	87.39	83.23	43.08	1616.63	186.67	147.31	105.11
obj_000014	hand	luxonis	SAM6D	circ_close	422.01	-55.73	-78.21	410.94	92.3	103.26	57.76	2015.34	148.34	150.66	109.92
obj_000014	hand	luxonis	SAM6D	zoom	107.5	33.73	-6.74	101.85	76.7	86.06	66.2	1407.08	164.66	162.64	173.14
obj_000014	hand	luxonis	SAM6D	all_motions	231.54	31.87	-33.83	226.82	85.73	90.13	54.32	1770.77	169.01	153.92	132.26
obj_000014	hand	realsense	SAM6D	circ_far	201.42	65.79	-52.14	183.09	78.83	69.01	40.83	1475.18	184.63	145.22	84.49
obj_000014	hand	realsense	SAM6D	circ_close	145.51	42.67	-52.57	128.79	76.5	62.95	37.53	1675.83	190.6	146.38	98.59
obj_000014	hand	realsense	SAM6D	zoom	55.27	18.99	51.5	6.48	99.73	86.31	60.25	529.31	165.98	149.73	158.65
obj_000014	hand	realsense	SAM6D	all_motions	131.1	46.61	-24.36	120.09	83.79	71.94	45.12	1420.79	182.53	148.17	115.25
obj_000014	approach	luxonis	SAM6D	circ_far	71.45	48.5	-6.45	52.07	59.67	44.1	26.58	778.58	228.92	167.07	103.79
obj_000014	approach	luxonis	SAM6D	circ_close	819.68	-37.6	-129.18	808.56	94.43	89.85	49.38	2251.08	114.8	81.71	79.87
obj_000014	approach	luxonis	SAM6D	zoom	51.61	-32.14	14.14	37.82	103.12	87.88	51.37	359.91	163.22	152.33	146.67
obj_000014	approach	luxonis	SAM6D	all_motions	271.15	1.82	-36.7	268.65	81.43	69.15	39.86	1797.02	192.62	157.04	115.87
obj_000014	approach	realsense	SAM6D	circ_far	6.93	1.4	3.23	-5.96	20.17	13.68	8.24	8.31	161.16	109.69	54.74
obj_000014	approach	realsense	SAM6D	circ_close	6.32	1.91	3.19	-5.1	62.05	43.47	24.02	19.99	242.81	170.29	86.46
obj_000014	approach	realsense	SAM6D	zoom	11.8	-5.52	10.08	-2.71	47.45	38.11	13.27	224.22	211.96	178.32	41.29
obj_000014	approach	realsense	SAM6D	all_motions	7	-0.29	5.04	-4.85	39.52	28.78	14.14	119.95	206.96	153.25	65.43
obj_000021	approach	luxonis	SAM6D	circ_far	49.28	28.34	23.45	32.78	55.47	59.62	55.86	652.86	124.97	128.12	145.21
obj_000021	approach	luxonis	SAM6D	circ_close	98.4	51.15	-49.73	67.77	80.76	80.09	86.2	400.07	122.63	123.64	122.64
obj_000021	approach	luxonis	SAM6D	zoom	29.1	19.34	1.39	21.69	84.8	92.65	67.33	370.31	163.98	156.06	142.08
obj_000021	approach	luxonis	SAM6D	all_motions	53.38	33.41	-5.66	41.25	71	74.54	68.52	526.61	139.92	139.36	141.85
obj_000021	approach	realsense	SAM6D	circ_far	30.23	-12.21	25.64	10.35	72.6	80.77	64.39	538.23	148.02	158.05	136.58
obj_000021	approach	realsense	SAM6D	circ_close	65.83	56.66	-21.62	25.59	73.23	77.26	67.33	513.41	162.56	163.39	155.56
obj_000021	approach	realsense	SAM6D	zoom	69.47	3.04	-19.55	66.59	92.02	89.44	75.77	932.98	121	141.3	138.05
obj_000021	approach	realsense	SAM6D	all_motions	33.53	13.43	-1.59	30.68	78.18	82.08	68.46	687.42	147.01	154.69	142.62
obj_000021	hand	luxonis	SAM6D	circ_far	28.96	14.53	2.78	24.9	58.62	54.5	58.61	393.67	140.63	134.39	149.22
obj_000021	hand	luxonis	SAM6D	circ_close	51.23	37.4	-4.56	34.71	75.13	79.9	61.38	396.72	140.81	156.68	120.93
obj_000021	hand	luxonis	SAM6D	zoom	32.87	15.5	14.47	25.11	93.63	88.38	71.73	419.78	93.04	71	120.75
obj_000021	hand	luxonis	SAM6D	all_motions	36.07	22.23	3.78	28.14	74.11	72.55	63.31	423.6	134.68	134.46	132.42
obj_000021	hand	realsense	SAM6D	circ_far	58.25	-0.23	56.31	-14.91	79.86	76.37	74.49	395.7	124.94	126.85	136.42
obj_000021	hand	realsense	SAM6D	circ_close	66.2	60.73	-8.27	25.02	86.13	90.41	77.72	392.25	144.79	146.43	121.84
obj_000021	hand	realsense	SAM6D	zoom	86.2	66.23	47.1	28.75	93.11	86.56	72.99	569.43	118.68	126.55	117.76
obj_000021	hand	realsense	SAM6D	all_motions	51.28	37.68	33.33	9.96	85.56	83.67	75.09	482.52	129.77	133.38	125.88
obj_000021	base	luxonis	SAM6D	circ_far	50.39	23.61	-8.89	43.62	12.39	13.8	8.99	793.78	111.85	106.33	61.64
obj_000021	base	luxonis	SAM6D	circ_close	14.49	-1.97	6.48	12.81	3.26	4.2	4.43	65.96	25.8	33.07	39.83
obj_000021	base	luxonis	SAM6D	zoom	17.25	-0.41	-1.37	17.19	12	15.77	9.38	131.39	93.15	116.09	62.99
obj_000021	base	luxonis	SAM6D	all_motions	27.95	8.84	-1.74	26.46	9.17	11	7.53	512.1	86.75	91.71	55.41
obj_000021	base	realsense	SAM6D	circ_far	4.59	-3.57	1.26	2.58	8.41	12.35	12.94	85.71	55.05	91.42	93.32
obj_000021	base	realsense	SAM6D	circ_close	5.81	-1.55	5.41	-1.47	5.01	5.91	6.18	65.31	47.82	54.96	57.45
obj_000021	base	realsense	SAM6D	zoom	11.4	-6.4	4.03	8.52	25.68	23.35	26.21	119.28	116.8	146	123.87
obj_000021	base	realsense	SAM6D	all_motions	5.85	-3.78	3.23	3.08	12.27	13.58	14.71	92.77	79.59	103.3	96.95

A.3.2. Versuch A2

object_name	occlusion	sensor	method	motion	AP_p	ap_x	ap_y	ap_z	ap_a	ap_b	ap_c	RP_i	RP_a	RP_b	RP_c
obj_000002	base	luxonis	FoundationPose	circ_far	12.56	-0.8	1.3	12.47	175.5	117.39	62.58	4.52	3.89	2.08	1.91
obj_000002	base	luxonis	FoundationPose	circ_close	90.68	-13.18	-1.43	89.7	63.7	75.41	106.16	42.42	50.64	28.16	38.83
obj_000002	base	luxonis	FoundationPose	zoom	16.92	-0.29	-0.03	16.92	177.12	140.72	39.16	8.54	6.62	2.99	2.41
obj_000002	base	luxonis	FoundationPose	all_motions	36.08	-4.21	0.16	35.83	143.87	111.68	68.72	85.69	155.53	76.53	79.85
obj_000002	base	realsense	FoundationPose	circ_far	2.75	1.1	2.5	-0.25	174.91	120.19	59.91	9.06	4.07	2.18	2.12
obj_000002	base	realsense	FoundationPose	circ_close	7.05	2.07	3.79	-5.58	169.78	120.86	60.02	4.66	3.15	2.91	2.72
obj_000002	base	realsense	FoundationPose	zoom	7.64	0.19	1.05	7.57	5.11	4.21	2.88	6.73	7.17	6.3	3.78
obj_000002	base	realsense	FoundationPose	all_motions	2.74	1.12	2.47	0.38	127.41	88.91	44.47	12.73	224.89	155.7	76.48
obj_000002	hand	luxonis	FoundationPose	circ_far	12.61	-0.34	0.3	12.6	36.84	28.45	17.2	4.85	139.92	97.81	55.61
obj_000002	hand	luxonis	FoundationPose	circ_close	96.66	3.37	-8.39	96.23	30.26	35.33	30.87	97.45	49.69	49.85	61.17
obj_000002	hand	luxonis	FoundationPose	zoom	44.99	-1.58	-4.13	44.77	30.28	25.56	16.36	200.09	57.49	58.91	25.29
obj_000002	hand	luxonis	FoundationPose	all_motions	45.45	0.39	-3.4	45.32	33.17	29.64	20.89	144.37	101.17	77.05	54.21

obj_000002	hand	realsense	FoundationPose	circ_far	4.72	1.29	4.49	-0.69	59.59	44.07	25.34	12.52	199.02	137.38	74.47
obj_000002	hand	realsense	FoundationPose	circ_close	21.36	8.28	13.12	14.68	18.85	13.27	18.82	57.2	54.13	34.03	60.38
obj_000002	hand	realsense	FoundationPose	zoom	15.49	-1.86	4.78	14.62	165.06	142.4	35.31	66.45	21.71	17.53	13.73
obj_000002	hand	realsense	FoundationPose	all_motions	10.78	2.43	7.02	7.81	76.39	61.78	26.17	57.42	217.56	178.77	62.16
obj_000002	approach	luxonis	FoundationPose	circ_far	13.48	-0.77	-1.06	13.42	13.73	11.96	7.95	7.1	29.43	25.62	14.06
obj_000002	approach	luxonis	FoundationPose	circ_close	306.13	-30.81	-88.15	291.54	98.07	111.88	55.35	366.08	102.75	94.44	74.38
obj_000002	approach	luxonis	FoundationPose	zoom	18.47	-0.14	-2.64	18.28	172.91	142.71	36.56	15.45	6.34	9.46	10.07
obj_000002	approach	luxonis	FoundationPose	all_motions	98.58	-9.22	-26.47	94.51	81.11	76.09	29.31	436.5	207.46	183.19	73.43
obj_000002	approach	realsense	FoundationPose	circ_far	4.49	1.34	4.29	-0.05	172.22	117.73	61.78	10.91	13.01	3	1.69
obj_000002	approach	realsense	FoundationPose	circ_close	6.67	2.5	5.38	-3.05	167.82	118.67	61.72	5.26	9.81	3.75	3.18
obj_000002	approach	realsense	FoundationPose	zoom	9.81	0.93	0.85	9.73	172.28	142.45	36.63	20.42	11.67	3.53	3.24
obj_000002	approach	realsense	FoundationPose	all_motions	4.35	1.56	3.67	1.74	170.97	124.71	54.94	18.42	13.18	32.79	33.75
obj_000002	base	luxonis	MegaPose	circ_far	7.07	-0.78	1.31	6.9	102.74	69.51	36.42	7.16	258.68	172.71	90.18
obj_000002	base	luxonis	MegaPose	circ_close	9.13	-0.05	3.35	8.49	96.63	68	36.23	4.25	248.46	172.3	91.81
obj_000002	base	luxonis	MegaPose	zoom	96.21	-0.6	8.18	95.86	125.28	91.18	132.54	123.79	219.72	155.28	99.54
obj_000002	base	luxonis	MegaPose	all_motions	31.72	-0.52	3.76	31.5	107.1	74.96	62.45	122.95	246.24	169.4	158.6
obj_000002	base	realsense	MegaPose	circ_far	3.16	0.64	1.32	2.8	67.03	46.39	23.73	7.96	252.99	173.8	85.94
obj_000002	base	realsense	MegaPose	circ_close	5.11	1.38	2.85	4.01	113.15	80.53	41.42	3.76	237.25	166.83	84.73
obj_000002	base	realsense	MegaPose	zoom	97.28	4.14	11.13	96.56	139.47	83.05	129.08	117.62	207.66	159.61	109.79
obj_000002	base	realsense	MegaPose	all_motions	28.97	1.8	4.41	28.57	99.56	65.86	57.25	123.51	253.08	175.06	162.37
obj_000002	hand	luxonis	MegaPose	circ_far	12.12	-1.22	-2.37	11.82	107.05	77.91	45.45	49.26	230.74	150.31	93.71
obj_000002	hand	luxonis	MegaPose	circ_close	13.74	0.11	2.39	13.53	116.33	89.01	52.76	32.82	202.06	140.45	98.16
obj_000002	hand	luxonis	MegaPose	zoom	155.19	5.52	-3.36	155.06	28.25	168.03	158.07	194.35	159.79	67.17	119.1
obj_000002	hand	luxonis	MegaPose	all_motions	51.2	0.99	-1.28	51.17	88.33	105.55	78.1	204.92	232.13	173.24	178.89
obj_000002	hand	realsense	MegaPose	circ_far	9.26	1.36	-2.68	8.76	87.76	62.23	39.09	42.16	235.11	153.1	100.3
obj_000002	hand	realsense	MegaPose	circ_close	6.05	1.89	2.1	5.35	123.93	88.34	46.7	14.88	121.73	147.22	76.38
obj_000002	hand	realsense	MegaPose	zoom	147.17	10.82	6.07	146.65	101.51	71.76	129.05	227.1	225.98	169.55	141.64
obj_000002	hand	realsense	MegaPose	all_motions	45.11	4.06	1.04	44.91	101.76	72.22	65.48	219.67	229.29	158.29	157.98
obj_000002	approach	luxonis	MegaPose	circ_far	6.37	-0.38	2.13	5.99	107.25	75.09	39.79	15.27	233.8	157.74	83.59
obj_000002	approach	luxonis	MegaPose	circ_close	14.46	-0.07	2.56	14.24	97.67	75.32	48.46	35.48	216.68	154.41	119.56
obj_000002	approach	luxonis	MegaPose	zoom	97.64	-0.56	8.29	97.28	35.73	154.79	153.02	107.87	187.34	99.72	111.15
obj_000002	approach	luxonis	MegaPose	all_motions	33.36	-0.34	3.92	33.13	85.1	96.78	73	119.49	234.13	177.76	179.17
obj_000002	approach	realsense	MegaPose	circ_far	4.76	0.46	1.58	4.47	109.68	79.39	45.77	37.85	225.99	157.5	105.11
obj_000002	approach	realsense	MegaPose	circ_close	5.39	1.31	3.54	3.85	113.17	81.32	42.71	7.74	228.64	161.22	83.01
obj_000002	approach	realsense	MegaPose	zoom	106.21	5.59	11.13	105.47	131.99	80.86	135.02	118.18	201.09	154.81	84.46
obj_000002	approach	realsense	MegaPose	all_motions	32.12	2.1	4.73	31.7	116.73	80.34	69.11	132.06	220.44	156.64	152.82
obj_000002	base	luxonis	SAM6D	circ_far	13.34	-1.45	0.07	13.26	64.98	45.79	24.6	6.02	236.71	160.01	82.37
obj_000002	base	luxonis	SAM6D	circ_close	170.13	-11.24	-26.75	167.63	67.48	78.85	72.43	591.44	147.27	87.97	111.21
obj_000002	base	luxonis	SAM6D	zoom	218.4	-19.62	-23.58	216.23	107.04	87.76	52.91	2115.45	202.61	152.67	121.18
obj_000002	base	luxonis	SAM6D	all_motions	108.75	-8.68	-13.58	107.55	75.65	65.63	45.67	1130.12	210.5	150.22	118.61
obj_000002	base	realsense	SAM6D	circ_far	3.49	1.28	3.25	-0.01	44.75	31.47	16.4	8.09	217.89	149.35	76.05
obj_000002	base	realsense	SAM6D	circ_close	21.35	4.53	15.3	14.18	82.29	80.24	83.92	32.18	202.45	161.72	180.49
obj_000002	base	realsense	SAM6D	zoom	23.32	12.98	-5.26	18.65	89.05	77.29	29.51	261.94	226.98	188.65	79.77
obj_000002	base	realsense	SAM6D	all_motions	11.1	5.12	4.61	8.7	66.55	56.88	39.1	141.66	222.8	176.99	144.97
obj_000002	hand	luxonis	SAM6D	circ_far	10.83	0.69	3.13	10.35	77.99	61.2	32.75	214.32	208.57	152.86	89.23
obj_000002	hand	luxonis	SAM6D	circ_close	493.9	-58.06	-108.5	478.33	105.57	114.65	42.89	1336.31	165.87	123.76	104.77
obj_000002	hand	luxonis	SAM6D	zoom	266.4	122.89	29.47	234.52	79.79	79.5	59.55	1689.34	144.3	125.64	158.6
obj_000002	hand	luxonis	SAM6D	all_motions	206.74	15.33	-22.55	204.94	86.49	81.53	42.67	1325.33	183.97	152.85	118.89
obj_000002	hand	realsense	SAM6D	circ_far	14.65	6.59	13.02	1.27	54.71	49.85	33.94	440.32	174.76	141.02	114.67
obj_000002	hand	realsense	SAM6D	circ_close	26.95	18.01	-18.05	8.71	73.52	68.38	53.15	242.67	178.64	149.89	141.25
obj_000002	hand	realsense	SAM6D	zoom	16.19	1.58	-1.08	16.08	91.79	82.19	30.53	195.46	176.13	149.94	59.44
obj_000002	hand	realsense	SAM6D	all_motions	11.2	8.42	0.52	7.36	70.02	63.8	38.38	344.82	180.89	150.49	114.17
obj_000002	approach	luxonis	SAM6D	circ_far	12.98	0.6	2.73	12.67	67.66	53.46	30.58	200.72	184.55	130.99	74.15
obj_000002	approach	luxonis	SAM6D	circ_close	289.56	-16.67	-57.99	283.2	86.95	87.19	58.34	1244.01	175.9	135.1	116.15
obj_000002	approach	luxonis	SAM6D	zoom	172.16	101.15	61.14	125.18	92.32	97.12	57.52	699.28	174.53	128.22	138.97
obj_000002	approach	luxonis	SAM6D	all_motions	121.61	22.62	1.15	119.48	79.73	74.71	45.66	921.86	181.17	143.04	113.71
obj_000002	approach	realsense	SAM6D	circ_far	4.78	1.81	4.42	0.02	46.62	35.02	19.19	10.22	194.34	136.45	72.05
obj_000002	approach	realsense	SAM6D	circ_close	32.41	12.94	22.18	19.77	89.34	85.12	83.85	158.4	167.21	133.34	159.58
obj_000002	approach	realsense	SAM6D	zoom	33.02	27.15	-4.31	18.3	105.87	92.45	31.21	232.94	196.12	161.47	57.53
obj_000002	approach	realsense	SAM6D	all_motions	17.48	11.88	7.15	10.64	74.95	64.97	41	161.4	201.78	163.04	131.28
obj_000006	approach	luxonis	FoundationPose	circ_far	14.22	-1.34	0.37	14.15	4.75	7.78	9.15	11.48	11.38	15.62	17.54
obj_000006	approach	luxonis	FoundationPose	circ_close	14.26	-0.5	3.11	13.91	5.35	7.42	8.65	9.97	11.38	17.66	20.09
obj_000006	approach	luxonis	FoundationPose	zoom	13.59	-0.99	-3.94	12.97	2.64	75.68	75.64	11.09	4.84	82.56	82.44

obj_000006	approach	luxonis	FoundationPose	all_motions	13.8	-1	-0.01	13.76	4.35	26.1	27.05	12.37	10.48	101.53	100.15
obj_000006	approach	realsense	FoundationPose	circ_far	0.77	-0.39	0.56	0.36	4.63	6.06	7.1	10.52	7.13	12.18	13.34
obj_000006	approach	realsense	FoundationPose	circ_close	2.35	0.6	1.92	1.2	3.33	4.56	4.71	7.22	5.21	7.32	7.09
obj_000006	approach	realsense	FoundationPose	zoom	8.39	-1.53	-2.76	7.77	5.07	97.51	97.62	7.76	3.93	124.49	124.18
obj_000006	approach	realsense	FoundationPose	all_motions	2.63	-0.41	0.06	2.59	4.38	30.25	30.79	12.7	6.19	138.86	138.17
obj_000006	base	luxonis	FoundationPose	circ_far	14.16	-1.5	-0.64	14.07	4.51	6.18	7.05	8.13	9.32	15.31	17.64
obj_000006	base	luxonis	FoundationPose	circ_close	13.26	-0.55	2.08	13.09	4.65	7.17	7.77	7.47	10.1	16.32	16.73
obj_000006	base	luxonis	FoundationPose	zoom	22.34	-2.71	-3.73	21.86	7.49	29.46	30.27	13.74	6.27	18.08	17.11
obj_000006	base	luxonis	FoundationPose	all_motions	15.99	-1.56	-0.7	15.9	5.36	12.78	13.56	15	9.62	34.72	35.17
obj_000006	base	realsense	FoundationPose	circ_far	0.62	-0.33	-0.43	-0.29	4.17	5.38	4.54	10.13	4.85	8.82	9.93
obj_000006	base	realsense	FoundationPose	circ_close	1.97	0.21	1.05	1.65	3.52	4.87	4.32	6.09	6.03	10.68	12.49
obj_000006	base	realsense	FoundationPose	zoom	10.34	-1.65	-4.76	9.03	5.97	62.72	63.04	8.87	2.51	8.55	8.54
obj_000006	base	realsense	FoundationPose	all_motions	3.06	-0.53	-1.18	2.77	4.47	20.67	20.23	14.31	5.52	77.42	78.94
obj_000006	hand	luxonis	FoundationPose	circ_far	16.02	-3.26	0.08	15.68	23.16	15.33	26.47	21.48	77.68	33.02	79.43
obj_000006	hand	luxonis	FoundationPose	circ_close	61.63	26.79	-9.44	54.69	85.27	106.57	56.6	123.66	59.06	53.35	103.12
obj_000006	hand	luxonis	FoundationPose	zoom	40.68	-10.33	-9.76	38.12	115.02	152.62	71.71	112.99	7.1	8.31	8.97
obj_000006	hand	luxonis	FoundationPose	all_motions	33.56	3.44	-5.32	32.96	65.9	78.74	47.38	114.89	133.68	181.27	95.94
obj_000006	hand	realsense	FoundationPose	circ_far	10.37	8.85	-4.54	2.92	72.16	59.06	82.21	32.67	57.15	59.64	24.82
obj_000006	hand	realsense	FoundationPose	circ_close	18.94	18.46	-1.26	-4.02	88.96	108.64	64.07	23.57	68.71	71.24	90.81
obj_000006	hand	realsense	FoundationPose	zoom	8.83	-0.1	-6.84	5.58	120.69	35.2	122.49	28.66	61.32	14.01	52.35
obj_000006	hand	realsense	FoundationPose	all_motions	10.19	9.1	-4.25	1.7	90.02	66.43	88.08	37.58	85.8	100.78	88.15
obj_000006	approach	luxonis	MegaPose	circ_far	23.43	-0.55	-1.05	-23.4	35.24	56.81	69.91	145.33	74.71	92.1	99.71
obj_000006	approach	luxonis	MegaPose	circ_close	7.44	1.5	-3.11	-6.59	39.19	45.66	54.62	94.92	114.05	94.09	90.5
obj_000006	approach	luxonis	MegaPose	zoom	25	7.76	-2.5	-23.63	54.05	23.36	58	75.14	144.28	63.7	134.7
obj_000006	approach	luxonis	MegaPose	all_motions	18.89	2.29	-2.03	-18.64	41.48	44.53	62.29	117.42	110.4	94.81	109.1
obj_000006	approach	realsense	MegaPose	circ_far	25.66	0.46	-4.84	-25.2	44.09	70.01	72.38	221.18	114	94.74	91.32
obj_000006	approach	realsense	MegaPose	circ_close	11.92	-0.72	-3.81	-11.28	47.6	59.97	55.31	141.25	147.86	140.84	119.15
obj_000006	approach	realsense	MegaPose	zoom	57.14	1.92	-4.85	-56.9	59.01	34.52	73.64	125.6	98.69	107.84	100.12
obj_000006	approach	realsense	MegaPose	all_motions	30.12	0.52	-4.55	-29.77	49.11	57.6	67.86	186.42	121.48	120.44	104.21
obj_000006	base	luxonis	MegaPose	circ_far	4.69	-2.46	-1.71	3.6	33.58	38.22	57.11	99.66	106.06	92.02	147.42
obj_000006	base	luxonis	MegaPose	circ_close	9.67	-0.83	-0.43	9.63	23.29	28.6	37.8	55.74	60.79	67.44	70.04
obj_000006	base	luxonis	MegaPose	zoom	3.79	-1.9	-3.28	0.02	9.78	39.15	38.57	38.11	25.26	155.16	136.13
obj_000006	base	luxonis	MegaPose	all_motions	5.05	-1.84	-1.77	4.36	24.17	35.71	46.54	78.06	83.64	107.54	128.89
obj_000006	base	realsense	MegaPose	circ_far	10.71	-1.48	-3.7	9.94	24.95	52.99	60.18	107.33	72.1	137.15	131.97
obj_000006	base	realsense	MegaPose	circ_close	5.47	-0.57	0.12	5.44	23.72	35.39	43.86	76.63	64.78	121.89	115.5
obj_000006	base	realsense	MegaPose	zoom	7.63	-2.13	-4.79	5.54	4.48	15.39	15.95	52.01	10.4	55.98	55.75
obj_000006	base	realsense	MegaPose	all_motions	8.14	-1.4	-2.91	7.47	19.09	37.86	43.63	88.91	64.9	124.33	123.36
obj_000006	hand	luxonis	MegaPose	circ_far	29.15	2.35	-2.89	-28.91	78.76	94.92	81.13	150.33	142.31	138.83	113.42
obj_000006	hand	luxonis	MegaPose	circ_close	25.56	1.12	-2.08	-25.45	69.98	81.99	67.83	106.34	124.86	131.77	90.94
obj_000006	hand	luxonis	MegaPose	zoom	63.91	-2.71	-9.3	-63.17	63.05	126.3	103.49	99.49	133.37	87.1	102.53
obj_000006	hand	luxonis	MegaPose	all_motions	37.48	0.63	-4.4	-37.21	71.98	99.72	83.38	134.72	135.5	134.09	111.32
obj_000006	hand	realsense	MegaPose	circ_far	37.15	-3.41	-3.14	-36.86	75.79	82.25	67.11	165.62	138.37	146.39	92.97
obj_000006	hand	realsense	MegaPose	circ_close	38.02	-2.23	-7.31	-37.24	80.61	75.31	74.52	98.54	128.22	120.98	92.78
obj_000006	hand	realsense	MegaPose	zoom	56.5	-6.92	-8.45	-55.43	32.27	114.58	106.15	101.51	50.42	139.28	116.87
obj_000006	hand	realsense	MegaPose	all_motions	42.59	-4.03	-5.75	-42.01	65.33	89.08	79.77	139.7	132	144.81	110.76
obj_000006	approach	luxonis	SAM6D	circ_far	69.58	33.04	57.5	-21.05	93.89	94.99	81.15	357.8	133.36	111.54	101.13
obj_000006	approach	luxonis	SAM6D	circ_close	83.69	10.03	-5.98	82.87	88.61	75.22	82.11	1460.96	125.64	119.98	131.34
obj_000006	approach	luxonis	SAM6D	zoom	95.12	59.22	-18.97	71.98	80.51	77.7	85.24	846.36	162.54	92.45	112.66
obj_000006	approach	luxonis	SAM6D	all_motions	59.28	32.9	8.56	48.56	87.53	81.88	82.85	1062.73	140.34	111.05	115.92
obj_000006	approach	realsense	SAM6D	circ_far	18.02	3.46	15.82	7.89	98	88.03	88.61	374.01	149.38	129.57	128.17
obj_000006	approach	realsense	SAM6D	circ_close	95.6	72.07	-33.38	53.21	69.97	81.8	90.97	963.38	132.68	154.07	143.03
obj_000006	approach	realsense	SAM6D	zoom	62.74	22.03	26.13	52.62	91.73	89.38	92.06	901.09	125.62	98.15	63.86
obj_000006	approach	realsense	SAM6D	all_motions	54.04	35.35	1.59	40.83	85.45	86.24	90.74	841.36	138.03	128.09	114.29
obj_000006	base	luxonis	SAM6D	circ_far	86.12	31.51	9.56	79.58	100.61	94.86	87.62	1212.38	155.29	156.88	117.69
obj_000006	base	luxonis	SAM6D	circ_close	56.61	52.45	19.73	7.98	78.54	82.49	71.42	297.78	133.62	136.86	116.39
obj_000006	base	luxonis	SAM6D	zoom	125.04	-2.66	-31.88	120.88	82.4	90.42	105.67	1128.48	195.94	83.48	72.63
obj_000006	base	luxonis	SAM6D	all_motions	74.29	28.17	0.62	68.73	88.15	89.57	87.73	1014.18	163.47	132.19	112.56
obj_000006	base	realsense	SAM6D	circ_far	36.76	17.16	10.1	30.9	98.1	88.09	89.63	798.02	164.73	149.48	127.03
obj_000006	base	realsense	SAM6D	circ_close	21.03	12.61	16.83	0.3	72.99	75.84	76.65	175.16	168.25	165.26	148.43
obj_000006	base	realsense	SAM6D	zoom	78.5	-36.89	42.97	54.37	97.56	86.28	92.13	946.45	238.86	76.45	79.33
obj_000006	base	realsense	SAM6D	all_motions	36.23	-0.52	22.05	28.75	90.36	83.85	86.46	744.41	192.46	137.12	122.78
obj_000006	hand	luxonis	SAM6D	circ_far	57.14	16.94	46.74	28.17	88.43	92.86	88.61	834.96	129.6	119.77	98.58
obj_000006	hand	luxonis	SAM6D	circ_close	125.3	122.12	17.05	22.29	91.34	85.67	88.06	479.8	110.43	106.35	94.39

obj_000006	hand	luxonis	SAM6D	zoom	135.45	66.69	-35.1	112.54	82.26	88.1	85.59	1197.5	130.81	62.96	58.44
obj_000006	hand	luxonis	SAM6D	all_motions	88.93	71.28	10.39	52.15	87.59	88.72	87.47	889.02	122.51	99.44	85.63
obj_000006	hand	realsense	SAM6D	circ_far	56.6	36.16	42.38	10.03	72.6	91.86	89.15	920.74	103.34	92.57	75.73
obj_000006	hand	realsense	SAM6D	circ_close	49.22	47.36	11.62	-6.69	104.25	89.91	83.29	336.23	90.4	139.42	103.35
obj_000006	hand	realsense	SAM6D	zoom	68.88	65.05	21.78	6.25	92.23	86.78	92.48	392.5	133.87	73.41	74.51
obj_000006	hand	realsense	SAM6D	all_motions	55.88	50.19	24.4	2.86	90.55	89.4	88.27	600.48	116.51	105.24	85.98
obj_000014	base	luxonis	FoundationPose	circ_far	5.21	-1.21	1.44	4.86	176.77	118.46	61.61	4.09	4.81	3.11	3.13
obj_000014	base	luxonis	FoundationPose	circ_close	170.37	83.18	31.28	145.35	103.69	124.91	57.06	47.93	117.54	52.13	71.91
obj_000014	base	luxonis	FoundationPose	zoom	14.84	0.94	-2.51	14.6	177.5	142.33	37.77	50.5	6.96	9.08	8.48
obj_000014	base	luxonis	FoundationPose	all_motions	54.04	23.58	8.92	47.8	156.01	126.79	53.84	165.88	117.88	40.89	48.8
obj_000014	base	realsense	FoundationPose	circ_far	6.26	0.2	1.21	-6.14	1.08	1.07	1.44	8.29	1.9	2.11	2.53
obj_000014	base	realsense	FoundationPose	circ_close	239.65	72.25	-66.6	218.58	95.54	72.44	121.3	103.82	84.26	128.36	39.84
obj_000014	base	realsense	FoundationPose	zoom	15.65	-0.56	-2.08	15.5	178.24	143.98	36.01	95.83	4.45	11.81	11.96
obj_000014	base	realsense	FoundationPose	all_motions	70.09	20.66	-19.13	64.19	76.24	60.31	45.2	257.33	226.32	190.66	153.09
obj_000014	hand	luxonis	FoundationPose	circ_far	7.55	-1.18	-0.81	7.42	175.59	120.71	59.37	7.92	3.45	8.13	8.02
obj_000014	hand	luxonis	FoundationPose	circ_close	626.08	28.52	-13.2	625.29	93.64	105.66	47.86	565.46	134.21	107.16	45.45
obj_000014	hand	luxonis	FoundationPose	zoom	273.28	-18.19	-66.64	264.41	175.49	166.29	15.05	439.73	13.37	25.98	22.69
obj_000014	hand	luxonis	FoundationPose	all_motions	255.35	2.73	-22.22	254.36	152.06	128.76	44.04	706.27	132.55	92.48	61.59
obj_000014	hand	realsense	FoundationPose	circ_far	5.76	1.04	0.54	-5.64	173.93	121.98	58.24	7.44	4.21	4.66	4.4
obj_000014	hand	realsense	FoundationPose	circ_close	331.15	52.43	-37.76	324.79	76.6	98.88	152.01	199.3	105.24	78.82	29.1
obj_000014	hand	realsense	FoundationPose	zoom	32.15	-1.47	4.72	31.77	169.1	145.32	34.99	210.68	18.75	15.35	20.99
obj_000014	hand	realsense	FoundationPose	all_motions	100.86	15.1	-9.31	99.29	144.7	121.69	78.83	356.97	142.02	67.43	143.95
obj_000014	approach	luxonis	FoundationPose	circ_far	4.98	-0.93	0.65	4.85	176.35	119.37	60.73	4.88	6.56	3.98	4.03
obj_000014	approach	luxonis	FoundationPose	circ_close	404.14	27.85	54.61	399.46	110.82	66.6	149.32	337.18	103.85	100.56	35.21
obj_000014	approach	luxonis	FoundationPose	zoom	37.56	4.82	-2.29	37.18	3.39	3.6	3.28	209.52	18.99	19.03	14.71
obj_000014	approach	luxonis	FoundationPose	all_motions	128.04	8.89	15.33	126.8	110.63	72.82	70.55	492.19	220.82	153.38	167.57
obj_000014	approach	realsense	FoundationPose	circ_far	6.37	0.52	1.34	-6.21	1.42	1.26	1.85	7.73	2.8	2.45	3.06
obj_000014	approach	realsense	FoundationPose	circ_close	155.03	-70.45	41.97	131.57	126.11	111.86	103.3	128.39	74.29	163.6	117.62
obj_000014	approach	realsense	FoundationPose	zoom	24.03	-0.43	-4.13	23.67	178.01	141.2	38.81	153.49	4.41	11.14	11
obj_000014	approach	realsense	FoundationPose	all_motions	47.45	-20.1	11.51	41.41	85.1	70.95	40.98	187.5	235.03	208.63	142.05
obj_000014	base	luxonis	MegaPose	circ_far	5.03	-1.06	1.98	-4.5	28.78	19.41	10.67	7.61	193.5	130.34	66.35
obj_000014	base	luxonis	MegaPose	circ_close	2.82	-0.39	2.55	1.13	24.32	17.14	9.4	5.53	176.97	123.03	62.4
obj_000014	base	luxonis	MegaPose	zoom	8.22	-1.1	-0.05	-8.15	46.78	36.84	10.9	14.26	236.36	187.23	49.68
obj_000014	base	luxonis	MegaPose	all_motions	4.28	-0.88	1.59	-3.88	32.38	23.49	10.37	13.38	201.89	147.01	60.66
obj_000014	base	realsense	MegaPose	circ_far	7.26	-0.23	1.42	-7.11	37.73	25.85	13.64	5.02	214.4	146.97	73.06
obj_000014	base	realsense	MegaPose	circ_close	5.14	0.81	2.05	-4.65	23.99	17.03	9.5	3.33	173.79	124.44	61.67
obj_000014	base	realsense	MegaPose	zoom	6.78	-1.5	-0.83	-6.56	122.5	98.33	25.97	7.95	249.24	200.59	50.12
obj_000014	base	realsense	MegaPose	all_motions	6.34	-0.28	0.99	-6.26	56.79	42.98	15.8	7.18	244.92	186.98	66.7
obj_000014	hand	luxonis	MegaPose	circ_far	6.23	-1.14	2.55	-5.56	35.72	24.7	13.33	11.85	210.44	140.62	69.63
obj_000014	hand	luxonis	MegaPose	circ_close	2.93	-0.58	2.81	-0.6	80.6	59.19	31.75	15.38	254.51	176.89	93.18
obj_000014	hand	luxonis	MegaPose	zoom	94.61	-1.29	-18.84	92.71	55.82	71.68	58.15	270.01	196.45	168.57	157.25
obj_000014	hand	luxonis	MegaPose	all_motions	22.77	-1.02	-3.18	22.52	54.04	47.34	30.77	260.2	225.9	169.93	119.16
obj_000014	hand	realsense	MegaPose	circ_far	6.72	0.16	1.08	-6.63	44.8	33.15	18.57	54.57	225.27	156.11	83.24
obj_000014	hand	realsense	MegaPose	circ_close	5.85	1.83	1.57	-5.33	65.05	48.01	25.58	12.98	245.33	172.91	82.87
obj_000014	hand	realsense	MegaPose	zoom	22.44	0.28	-0.99	22.41	56.66	49.61	23.41	302.28	232.38	188.87	108.37
obj_000014	hand	realsense	MegaPose	all_motions	1.88	0.67	0.66	1.62	53.83	41.88	21.89	174.94	232.69	170.58	90.38
obj_000014	approach	luxonis	MegaPose	circ_far	4.72	-0.97	1.79	-4.26	19.8	13.54	7.87	6.77	163.57	108.65	56.11
obj_000014	approach	luxonis	MegaPose	circ_close	2.66	-0.57	2.25	1.31	28.84	20.8	11.09	5.93	189.88	132.67	66.72
obj_000014	approach	luxonis	MegaPose	zoom	8.52	-1.42	-0.14	-8.4	46.61	36.92	10.72	14.21	236.7	187.1	49.98
obj_000014	approach	luxonis	MegaPose	all_motions	4.15	-0.98	1.4	-3.79	29.67	21.97	9.57	12.97	194.49	142.31	57.57
obj_000014	approach	realsense	MegaPose	circ_far	7.81	-0.13	1.45	-7.67	19.69	13.87	7.86	5.81	161.29	110.22	54.37
obj_000014	approach	realsense	MegaPose	circ_close	5.43	0.56	1.9	-5.06	14.97	10.85	6.36	4.25	139.68	99.39	48.75
obj_000014	approach	realsense	MegaPose	zoom	6.61	-1.63	-0.82	-6.36	102.27	82.18	21.79	7.84	266.26	213.84	53.58
obj_000014	approach	realsense	MegaPose	all_motions	6.64	-0.34	0.97	-6.57	40.74	31.53	11.21	7.78	220.27	170.05	55.73
obj_000014	base	luxonis	SAM6D	circ_far	7.46	-1.46	1.33	7.19	14.5	10.43	5.98	6.79	133.64	89.75	46.41
obj_000014	base	luxonis	SAM6D	circ_close	833.3	-133.71	-170.55	804.63	87.16	92.09	37.61	3524.69	152.85	129.38	48.69
obj_000014	base	luxonis	SAM6D	zoom	227.46	-37.29	27.53	222.68	90.16	83.35	56.25	1725.26	155.05	136.18	151.98
obj_000014	base	luxonis	SAM6D	all_motions	301.87	-49.21	-41.39	294.94	55.6	53.4	28.48	2352.45	181.82	163.27	108.38
obj_000014	base	realsense	SAM6D	circ_far	5.35	0.86	1.98	-4.9	13.69	9.07	5.55	8.85	135.35	91.42	46.25
obj_000014	base	realsense	SAM6D	circ_close	426.82	-14.31	-138.77	403.38	91.47	95.95	55.34	1630.11	166.34	135.11	125.47
obj_000014	base	realsense	SAM6D	zoom	55.35	37.98	24.24	32.15	51.01	51.21	22.26	729.31	192.82	180.53	77.59
obj_000014	base	realsense	SAM6D	all_motions	124.31	6.74	-31.52	120.06	45.77	45.03	24.12	1306.97	187.65	171.39	103.84
obj_000014	hand	luxonis	SAM6D	circ_far	134.98	17.51	-25.38	131.41	98.38	78.98	44.34	1547.15	213.83	152.32	92.27

obj_000014	hand	luxonis	SAM6D	circ_close	602.98	-60.94	-136.03	584.27	93.28	95.12	55.5	2922.88	136.72	115.22	103.21
obj_000014	hand	luxonis	SAM6D	zoom	204.54	-69.04	-33.33	189.63	85.05	90.03	71.13	1170.11	179.75	162.02	150.71
obj_000014	hand	luxonis	SAM6D	all_motions	288.63	-28.56	-60.23	280.83	93.35	86.67	54.72	2081.9	184.23	145.56	117.21
obj_000014	hand	realsense	SAM6D	circ_far	125.8	20.36	-27.85	120.98	80.12	66.58	40.5	1353.44	226.15	169.44	107.78
obj_000014	hand	realsense	SAM6D	circ_close	368.71	37.27	-103.72	351.85	101.23	96.54	56.33	1577.49	164.21	147.9	113.16
obj_000014	hand	realsense	SAM6D	zoom	150.58	19.2	-0.67	149.34	69.07	90.92	74.33	948.56	197.02	120.69	140.57
obj_000014	hand	realsense	SAM6D	all_motions	207.07	25.37	-44.73	200.58	83.93	82.16	54.07	1428.58	202.96	155.97	124.44
obj_000014	approach	luxonis	SAM6D	circ_far	8.44	-1.83	5.78	5.87	33.68	25.37	14	128.95	192.87	134.01	68.88
obj_000014	approach	luxonis	SAM6D	circ_close	762.59	-74.58	-85.31	754.12	83.64	87.32	55.02	1885.38	129.5	97.05	110.98
obj_000014	approach	luxonis	SAM6D	zoom	203.53	-47.94	5.79	197.72	95.01	91.52	50.11	994.11	155.1	152.45	134.57
obj_000014	approach	luxonis	SAM6D	all_motions	271.76	-34.9	-19.84	268.78	64.5	60.88	35.41	1580.02	185.86	161.06	117.06
obj_000014	approach	realsense	SAM6D	circ_far	10.14	5.47	6.63	-5.37	24.84	17.19	11.09	116.35	174.66	121.82	71.12
obj_000014	approach	realsense	SAM6D	circ_close	110.86	32.97	-3.55	105.78	85.98	86.37	60.37	1128.11	154.95	131.57	118.25
obj_000014	approach	realsense	SAM6D	zoom	85.84	45.13	40.61	60.68	61.42	58.58	36.14	732.47	199.65	170.57	123.67
obj_000014	approach	realsense	SAM6D	all_motions	50.58	23.54	12.71	42.93	51.47	47.33	31.4	753.41	191.91	163.89	118.01
obj_000021	approach	luxonis	FoundationPose	circ_far	7.78	0.7	1.55	7.59	1.89	1.64	1.45	14.46	2.84	2.31	2.37
obj_000021	approach	luxonis	FoundationPose	circ_close	9.58	1.07	3.54	8.84	2.67	2.71	2.64	9.64	3.16	3.2	3.97
obj_000021	approach	luxonis	FoundationPose	zoom	11.76	-0.51	0.49	11.74	0.9	1.15	1.13	10.35	1.55	1.94	2.05
obj_000021	approach	luxonis	FoundationPose	all_motions	9.27	0.48	1.83	9.07	1.84	1.82	1.7	13.27	3.31	3.09	3.37
obj_000021	approach	realsense	FoundationPose	circ_far	5.77	0.86	2.33	-5.21	1.32	1.36	1.8	7.26	2.21	2.12	2.26
obj_000021	approach	realsense	FoundationPose	circ_close	5.35	2.06	3.83	-3.13	1.26	1.63	1.88	6.25	2.33	2.69	2.81
obj_000021	approach	realsense	FoundationPose	zoom	3.07	-0.44	1.23	2.78	1.69	2.08	2.47	7.28	2.41	2.54	3.12
obj_000021	approach	realsense	FoundationPose	all_motions	3.59	0.85	2.46	-2.47	1.4	1.63	2	10.3	2.34	2.55	2.78
obj_000021	hand	luxonis	FoundationPose	circ_far	6.43	-0.23	2.3	6	2.84	3.04	3.04	16.67	5.25	5.46	4.81
obj_000021	hand	luxonis	FoundationPose	circ_close	10.3	6.15	6.91	4.52	8.44	9.09	6.32	17.69	19.93	16.62	8.58
obj_000021	hand	luxonis	FoundationPose	zoom	6.12	-2.13	1.5	5.53	1.66	1.31	1.78	13.23	3.03	4.63	3.64
obj_000021	hand	luxonis	FoundationPose	all_motions	6.5	1.05	3.38	5.45	4.09	4.27	3.62	19.59	13.93	13.51	7.91
obj_000021	hand	realsense	FoundationPose	circ_far	5.27	0	1.31	-5.1	2.67	3.13	3.58	11.18	5.76	5.26	5.12
obj_000021	hand	realsense	FoundationPose	circ_close	6.87	0.18	2.38	-6.44	2.82	2.67	2.86	10.23	6.13	4.9	5.36
obj_000021	hand	realsense	FoundationPose	zoom	2.6	-1	1.64	-1.75	1.79	2.13	2.45	9.31	1.85	1.25	1.89
obj_000021	hand	realsense	FoundationPose	all_motions	4.89	-0.22	1.71	-4.58	2.48	2.73	3.07	10.1	5.25	4.56	4.76
obj_000021	base	luxonis	FoundationPose	circ_far	11.01	0.34	2.33	10.76	1.67	2.02	1.79	10.84	1.92	1.56	2.25
obj_000021	base	luxonis	FoundationPose	circ_close	11.1	1.45	4.5	10.05	2.33	2.22	2.16	6.19	2.41	1.67	2.6
obj_000021	base	luxonis	FoundationPose	zoom	14.39	-0.07	1.61	14.3	0.93	0.67	1.04	11.17	1.36	0.97	1.48
obj_000021	base	luxonis	FoundationPose	all_motions	11.85	0.55	2.75	11.51	1.66	1.71	1.69	11.54	2.49	2.41	2.52
obj_000021	base	realsense	FoundationPose	circ_far	3.62	1.25	2.27	-2.53	1.22	1.2	1.5	6.95	1.64	2.28	2.54
obj_000021	base	realsense	FoundationPose	circ_close	4.6	2.24	3.37	-2.19	1.19	1.09	1.51	4.5	2.27	1.8	2.04
obj_000021	base	realsense	FoundationPose	zoom	5.55	-0.09	0.96	5.47	1.41	0.9	1.66	8.66	2.47	1.62	2.61
obj_000021	base	realsense	FoundationPose	all_motions	2.53	1.17	2.23	-0.26	1.27	1.09	1.55	10.97	2.08	2	2.42
obj_000021	approach	luxonis	MegaPose	circ_far	20.98	1.42	0.81	20.91	36.07	35.29	37.25	312.58	144.99	128.15	134.56
obj_000021	approach	luxonis	MegaPose	circ_close	11.54	3.71	2.59	10.62	30.26	34.81	41.8	230.63	123.46	124.11	147.86
obj_000021	approach	luxonis	MegaPose	zoom	12.06	0.86	1.35	11.96	4.99	3.05	6.24	154.72	33.32	8.09	31.46
obj_000021	approach	luxonis	MegaPose	all_motions	15.72	1.93	1.46	15.53	25.97	26.41	30.14	257.63	123.77	115.66	127.65
obj_000021	approach	realsense	MegaPose	circ_far	22.12	4.27	2.71	21.54	32.12	33.52	41.07	234.32	144.51	131.27	160.19
obj_000021	approach	realsense	MegaPose	circ_close	11.23	3.91	6.36	-8.39	38.95	36.2	44.67	182.64	161.11	121.84	141.48
obj_000021	approach	realsense	MegaPose	zoom	74.37	4.98	3.97	74.1	28.85	29.3	28.45	285.05	80.18	125.05	82.2
obj_000021	approach	realsense	MegaPose	all_motions	27.82	4.36	4.09	27.17	33.18	33.15	38.7	256.73	135.28	126.25	138.12
obj_000021	hand	luxonis	MegaPose	circ_far	29.42	5.8	2.41	-28.74	56.84	59.97	65	140.52	170.1	155.89	158.85
obj_000021	hand	luxonis	MegaPose	circ_close	49.24	8.59	-1.24	-48.47	82.32	58.38	72.6	142.87	189.85	139.59	151.2
obj_000021	hand	luxonis	MegaPose	zoom	133.66	6.07	4.62	133.44	110.09	87.6	99.32	295.76	121.91	111.68	100.05
obj_000021	hand	luxonis	MegaPose	all_motions	12.39	6.66	1.99	10.26	78.67	67.11	76.57	323.44	176.29	144.42	148.18
obj_000021	hand	realsense	MegaPose	circ_far	25.32	4.23	1.37	-24.92	61.95	55.07	62.28	208.66	184.44	166.3	175.36
obj_000021	hand	realsense	MegaPose	circ_close	48.79	2.79	-3.13	-48.61	75.01	57.78	81.14	190.29	171.05	144.99	155.51
obj_000021	hand	realsense	MegaPose	zoom	19.59	6.5	-0.3	18.48	112.72	118.26	47.78	187.37	182.56	183.37	120.81
obj_000021	hand	realsense	MegaPose	all_motions	20.43	4.43	-0.37	-19.94	79.47	72.99	63.75	214.29	189.65	184.05	159.86
obj_000021	base	luxonis	MegaPose	circ_far	3.36	1.3	1.71	-2.58	3.39	3.12	4.36	32.41	21.6	9.52	20.07
obj_000021	base	luxonis	MegaPose	circ_close	5.72	1.27	3.17	4.59	3.23	7.82	8.51	95.26	9.44	67.87	68.72
obj_000021	base	luxonis	MegaPose	zoom	14.74	0.97	-1.78	-14.6	1.52	5.62	5.65	51.12	3.62	12.79	13.16
obj_000021	base	luxonis	MegaPose	all_motions	4.14	1.2	1.19	-3.78	2.84	5.13	5.89	66.41	15.5	37.47	39.61
obj_000021	base	realsense	MegaPose	circ_far	3.22	2.49	2.01	-0.37	7.09	4.92	8.52	54.74	44.45	17.44	42.41
obj_000021	base	realsense	MegaPose	circ_close	6.42	1.64	0.78	-6.15	2.36	3.29	4.3	18.55	6.34	7.99	8.54
obj_000021	base	realsense	MegaPose	zoom	20.67	2.57	-3.25	-20.25	9.96	9.06	13.81	46.47	45.82	16.34	42.48
obj_000021	base	realsense	MegaPose	all_motions	7.76	2.27	0.23	-7.42	6.51	5.58	8.74	51.97	38.84	16.34	37.44

obj_000021	approach	luxonis	SAM6D	circ_far	40.35	29.3	24.29	13.41	53.96	54.41	39.12	405.66	159.07	164.57	122.25
obj_000021	approach	luxonis	SAM6D	circ_close	64.92	56.46	3.77	31.82	63.47	65.01	57.59	381.71	166.79	171.3	139.74
obj_000021	approach	luxonis	SAM6D	zoom	33.68	21.07	16.23	20.66	78.63	107.98	70.13	305.36	161.7	162.13	151
obj_000021	approach	luxonis	SAM6D	all_motions	45.06	36.48	14.85	21.89	64.23	73.2	54.31	382.94	163.67	177.76	140.81
obj_000021	approach	realsense	SAM6D	circ_far	35.67	14.89	31.42	7.97	54.71	53.52	45.82	497.79	160.83	171.31	138.2
obj_000021	approach	realsense	SAM6D	circ_close	13.43	3.26	5.99	11.58	49.03	52.42	51.91	471.33	174.49	172.34	173.43
obj_000021	approach	realsense	SAM6D	zoom	84.25	-32.02	29.89	71.97	52.44	57.17	58.1	1040.68	127.15	164.44	155.09
obj_000021	approach	realsense	SAM6D	all_motions	36.66	-3.36	22.47	28.77	52.12	54.27	51.61	721.31	155.04	168.14	155.11
obj_000021	hand	luxonis	SAM6D	circ_far	83.63	72.9	39.6	10.57	83.14	81.54	76.42	403.75	129.11	140.84	133.7
obj_000021	hand	luxonis	SAM6D	circ_close	62.95	3.67	-17.5	60.36	70.34	79.62	59.61	517.52	151.47	153.14	99.39
obj_000021	hand	luxonis	SAM6D	zoom	58.5	29.31	1.52	50.6	82.16	83.3	74.28	453.11	131.91	118.32	150.48
obj_000021	hand	luxonis	SAM6D	all_motions	54.91	37.89	10.06	38.44	78.86	81.49	70.54	478.5	137.15	137.24	130.69
obj_000021	hand	realsense	SAM6D	circ_far	69.88	-12	67.02	-15.72	83.46	71.2	73.88	382.54	132.83	139.63	129.42
obj_000021	hand	realsense	SAM6D	circ_close	50.03	20.39	-28.7	35.54	81.65	81.48	85.92	434.96	131.19	134.69	129.66
obj_000021	hand	realsense	SAM6D	zoom	77.57	17.96	27.26	70.37	87.82	87.58	84.7	946.97	81.37	134.74	133.65
obj_000021	hand	realsense	SAM6D	all_motions	36.73	6.42	26.58	24.52	84.16	79.02	80.64	650.17	119.08	137.22	130.74
obj_000021	base	luxonis	SAM6D	circ_far	15.89	-2.37	1.5	15.64	4.99	5.74	6.61	117.22	44.98	48.48	58.44
obj_000021	base	luxonis	SAM6D	circ_close	11	-0.3	4.19	10.17	2.09	2.14	1.98	4.39	2.55	2.45	2.41
obj_000021	base	luxonis	SAM6D	zoom	13.61	0.47	3.57	13.13	4.99	7.11	8.77	28.18	37.7	53.58	60.41
obj_000021	base	luxonis	SAM6D	all_motions	13.75	-1.01	2.82	13.42	4.19	5.12	5.92	79.87	35.97	43.02	50.58
obj_000021	base	realsense	SAM6D	circ_far	2.62	-0.01	1.76	-1.95	1.62	1.61	2.07	8.13	2.43	3.95	3.88
obj_000021	base	realsense	SAM6D	circ_close	4.28	0.94	3.18	-2.71	1.03	1.36	1.11	4.84	2.18	2.35	2.38
obj_000021	base	realsense	SAM6D	zoom	8.34	-1.77	-3.62	7.31	6.76	7.16	7.55	127.05	58.38	63.38	78.24
obj_000021	base	realsense	SAM6D	all_motions	0.81	-0.22	0.68	0.39	2.87	3.07	3.31	68.9	31.25	33.91	41.51

A.3.3. Versuch B1

object_name	occlusion	sensor	method	motion	AP_p	ap_x	ap_y	ap_z	ap_a	ap_b	ap_c	RP_i	RP_a	RP_b	RP_c
obj_000014	base	luxonis	FoundationPose	circ_far	5.87	-0.03	0.83	5.81	176.01	118.24	61.8	30.05	4.99	5.01	4.76
obj_000014	base	luxonis	FoundationPose	circ_close	270.61	83.93	-11.31	257.02	77.67	106.09	58.59	443.2	106.24	38.29	68.29
obj_000014	base	luxonis	FoundationPose	zoom	46.26	1.78	-2.32	46.17	176.93	141.56	38.48	73.65	9.79	12.24	12.88
obj_000014	base	luxonis	FoundationPose	all_motions	91.63	24.37	-3.48	88.27	148.26	121.1	54.57	425.68	145.35	45.84	47.52
obj_000014	base	realsense	FoundationPose	circ_far	5.63	-1.36	0.86	-5.39	1.42	1.65	1.22	5.74	1.05	1.88	2.22
obj_000014	base	realsense	FoundationPose	circ_close	273.79	72.13	-93.71	246.93	95.03	42.39	109	185.77	133.83	88.67	85.39
obj_000014	base	realsense	FoundationPose	zoom	2.72	-0.02	-2.72	0	173.64	142.93	38.85	46.39	35.94	6.05	14.11
obj_000014	base	realsense	FoundationPose	all_motions	78.67	20.7	-28.02	70.54	75.4	51.69	43.16	320.83	227.06	180.52	143.86
obj_000002	base	luxonis	FoundationPose	circ_far	11.8	-0.36	1.36	11.71	172.9	117.02	62.33	32.85	19.02	9.25	6.15
obj_000002	base	luxonis	FoundationPose	circ_close	123.4	-58.64	49.32	96.73	65.98	69.86	86.83	86.31	64.42	51.1	34.08
obj_000002	base	luxonis	FoundationPose	zoom	45.93	-1.46	-0.47	45.9	2.31	4.73	4.33	22.69	3.73	5.11	4.38
obj_000002	base	luxonis	FoundationPose	all_motions	50.33	-17.15	14.44	45.06	96.33	73.19	53.52	139.26	221.12	141.45	97.05
obj_000002	base	realsense	FoundationPose	circ_far	2.14	-0.25	2.1	0.36	174.67	120.23	59.87	3.82	3.99	1.51	1.55
obj_000002	base	realsense	FoundationPose	circ_close	6.23	1.13	3.65	-4.93	169.57	121.13	59.76	6.16	2.47	3.25	2.94
obj_000002	base	realsense	FoundationPose	zoom	2.05	-0.03	0.39	-2.02	9.36	8.12	4.73	9.32	9.18	8.35	3.98
obj_000002	base	realsense	FoundationPose	all_motions	2.79	0.21	2.09	-1.83	128.8	90.4	45.03	8.05	217.44	149.76	73.38
obj_000006	base	luxonis	FoundationPose	circ_far	14.1	-1.02	-1.7	13.96	7.67	9	12.01	26.23	17.72	14.21	19.64
obj_000006	base	luxonis	FoundationPose	circ_close	13.04	-0.31	1.39	12.96	5.95	9.48	11.05	8.43	11.39	20.2	22.31
obj_000006	base	luxonis	FoundationPose	zoom	48.93	-3.44	-2.29	48.76	5.58	16.65	17.67	20.93	7.55	31.47	30.71
obj_000006	base	luxonis	FoundationPose	all_motions	23.21	-1.48	-0.99	23.15	6.61	11.22	13.28	42.71	14.12	23.92	25.15
obj_000006	base	realsense	FoundationPose	circ_far	1.38	-1.38	0.01	0.02	2.45	3.79	3.23	6.59	3.86	8.68	9.41
obj_000006	base	realsense	FoundationPose	circ_close	2.19	-0.3	1.66	1.39	2.12	3.16	3.31	5.18	3.74	8.68	8.91
obj_000006	base	realsense	FoundationPose	zoom	5.02	-2.51	-4.32	0.48	5.97	16.38	17.59	9.77	7.18	23.74	22.78
obj_000006	base	realsense	FoundationPose	all_motions	1.63	-1.38	-0.69	0.54	3.31	7.03	7.15	9.49	6.96	22.39	23.83
obj_000021	base	luxonis	FoundationPose	circ_far	11.09	-0.19	2.21	10.87	1.49	2.06	1.81	17.81	1.61	2.51	2.74
obj_000021	base	luxonis	FoundationPose	circ_close	10.01	0.82	4.33	8.99	1.81	1.58	1.42	7.77	1.81	1.64	1.72
obj_000021	base	luxonis	FoundationPose	zoom	43.4	-0.85	3.32	43.26	1.23	0.83	1.42	22.06	1.39	1.31	1.65
obj_000021	base	luxonis	FoundationPose	all_motions	19.31	-0.08	3.11	19.05	1.51	1.59	1.59	39.29	1.74	2.52	2.29
obj_000021	base	realsense	FoundationPose	circ_far	2.86	-0.5	2.02	-1.97	1.75	2.01	1.71	5.54	1.27	3.1	3.31
obj_000021	base	realsense	FoundationPose	circ_close	3.62	0.89	3.18	-1.47	2.03	1.81	1.54	4.78	1.76	1.54	2
obj_000021	base	realsense	FoundationPose	zoom	4.25	-1.4	0.55	-3.97	1.11	0.81	1.25	6.04	1.57	1.03	1.81

obj_000021	base	realsense	FoundationPose	all_motions	3.09	-0.35	1.95	-2.37	1.66	1.63	1.54	6.68	1.84	2.74	2.68
------------	------	-----------	----------------	-------------	------	-------	------	-------	------	------	------	------	------	------	------

A.3.4. Versuch B2

object_name	occlusion	sensor	method	motion	AP_p	ap_x	ap_y	ap_z	ap_a	ap_b	ap_c	RP_i	RP_a	RP_b	RP_c
obj_000014	base	luxonis	FoundationPose	circ_far	5.61	-1.61	1.19	5.24	176.67	118.46	61.63	4.44	4.68	3.49	3.45
obj_000014	base	luxonis	FoundationPose	circ_close	381.56	98.25	-25.55	367.81	84.23	121.34	83.3	194.45	61.78	81.74	54.63
obj_000014	base	luxonis	FoundationPose	zoom	21.93	2.44	-4.42	21.34	2.6	2.48	3.25	115.23	8.31	13.08	13.04
obj_000014	base	luxonis	FoundationPose	all_motions	116.87	28.02	-7.97	113.18	103.23	87.95	52.05	420.07	219.92	162.39	97.93
obj_000014	base	realsense	FoundationPose	circ_far	5.55	-1.19	0.8	-5.37	1.31	1.55	1.22	4.68	1.19	1.86	2.26
obj_000014	base	realsense	FoundationPose	circ_close	279.51	70.97	-79.34	258.45	103.25	60.77	120.97	170.7	106.81	150.93	54.18
obj_000014	base	realsense	FoundationPose	zoom	3.01	-0.22	-2.42	1.79	173.81	142.78	38.88	47.74	35.31	4.94	13.53
obj_000014	base	realsense	FoundationPose	all_motions	77.55	19.56	-22.83	71.49	77.11	56.72	45.45	323.33	226.46	191.93	153.14
obj_000002	base	luxonis	FoundationPose	circ_far	12.38	-0.47	1.68	12.26	175.73	117.45	62.54	4.59	3.28	2.04	1.93
obj_000002	base	luxonis	FoundationPose	circ_close	128.54	-66.75	44.82	100.29	63.79	68.47	85.02	86.33	60.58	48.41	39.76
obj_000002	base	luxonis	FoundationPose	zoom	17.28	-0.12	0.73	17.27	177.16	140.84	39.06	8.95	5.57	2.98	2.52
obj_000002	base	luxonis	FoundationPose	all_motions	45.24	-19.19	13.67	38.62	144.33	109.89	62.55	150.92	155.81	87.42	55.64
obj_000002	base	realsense	FoundationPose	circ_far	2.88	0.22	2.86	-0.18	174.39	120.19	59.87	3.44	4.32	1.67	1.7
obj_000002	base	realsense	FoundationPose	circ_close	7.2	1.56	4.57	-5.33	3.37	3.48	2.64	7.32	7.12	5.37	3.34
obj_000002	base	realsense	FoundationPose	zoom	1.6	0.24	1.37	-0.79	6.8	5.69	3.76	9.64	6.93	6.2	3.25
obj_000002	base	realsense	FoundationPose	all_motions	3.51	0.61	2.94	-1.81	80.31	55.95	28.38	8.9	252.92	172.72	84.67
obj_000006	base	luxonis	FoundationPose	circ_far	13.78	-1.43	-0.52	13.7	3.99	6	6.59	8.13	7.82	15.35	16.91
obj_000006	base	luxonis	FoundationPose	circ_close	13.32	-0.59	2.24	13.12	3.63	5.75	6.27	7.64	7.84	12.21	12.99
obj_000006	base	luxonis	FoundationPose	zoom	22.08	-2.61	-3.1	21.71	5.85	29.84	30.29	12.77	6.52	13.99	13.55
obj_000006	base	luxonis	FoundationPose	all_motions	15.77	-1.51	-0.43	15.69	4.39	12.35	12.88	14.27	7.95	34.88	35.11
obj_000006	base	realsense	FoundationPose	circ_far	1.09	-1.08	-0.1	-0.07	2.73	3.93	3.21	5.79	4.35	8.22	8.68
obj_000006	base	realsense	FoundationPose	circ_close	2.1	-0.09	1.49	1.48	2.33	3.65	3.78	5.3	5.26	10.65	11.63
obj_000006	base	realsense	FoundationPose	zoom	5.27	-1.56	-5	0.58	6.28	64.14	64.51	6.87	2.56	13.06	13
obj_000006	base	realsense	FoundationPose	all_motions	1.45	-0.93	-0.97	0.55	3.57	20.12	19.93	8.5	6.53	81.15	82.21
obj_000021	base	luxonis	FoundationPose	circ_far	11.49	-0.25	1.7	11.36	1.62	2.12	1.73	9.37	1.67	2.12	2.69
obj_000021	base	luxonis	FoundationPose	circ_close	11.18	0.85	3.95	10.42	2.21	2.21	2.04	6.17	2.55	1.77	2.91
obj_000021	base	luxonis	FoundationPose	zoom	15.28	-0.81	0.97	15.23	0.74	0.69	0.81	11.41	1.12	1.03	1.43
obj_000021	base	luxonis	FoundationPose	all_motions	12.32	-0.09	2.14	12.13	1.55	1.76	1.57	11.34	2.49	2.65	2.87
obj_000021	base	realsense	FoundationPose	circ_far	2.09	-0.33	1.38	-1.53	1.74	2.02	1.7	4.36	1.25	3.06	3.44
obj_000021	base	realsense	FoundationPose	circ_close	3.04	1	2.64	-1.12	2.08	1.84	1.57	4.29	1.57	1.64	2.07
obj_000021	base	realsense	FoundationPose	zoom	2.33	-1.19	-0.27	-1.99	1.2	0.87	1.34	6.02	1.87	1.15	2.21
obj_000021	base	realsense	FoundationPose	all_motions	2.02	-0.18	1.29	-1.54	1.69	1.66	1.56	5.86	1.82	2.72	2.82

A.3.5. Versuch C2

object_name	occlusion	sensor	method	motion	AP_p	ap_x	ap_y	ap_z	ap_a	ap_b	ap_c	RP_i	RP_a	RP_b	RP_c
obj_000014	base	luxonis	FoundationPose	circ_far	5.53	-1.75	0.94	5.16	176.66	118.43	61.65	4.48	4.84	3.4	3.37
obj_000014	base	luxonis	FoundationPose	circ_close	432.56	48.51	37.2	428.22	46.82	61.75	51.96	244.42	41.4	36.31	34.41
obj_000014	base	luxonis	FoundationPose	zoom	14.22	-0.88	-2.66	13.94	177.64	142.03	38.03	33.3	5.16	6.68	6.32
obj_000014	base	luxonis	FoundationPose	all_motions	128.3	12.7	10.24	127.25	140.18	108.73	52.56	470.44	177.78	95.46	34.58
obj_000014	base	realsense	FoundationPose	circ_far	5.45	-0.83	0.54	-5.36	1.09	1.3	1.25	7.44	1.35	1.94	2.46
obj_000014	base	realsense	FoundationPose	circ_close	3.02	-0.13	2.78	1.17	170	118.53	62.44	13.26	4.93	7.54	7.63
obj_000014	base	realsense	FoundationPose	zoom	6.19	-1.62	-2.51	5.42	1.64	1.87	1.23	35.17	6.3	7.43	4.76
obj_000014	base	realsense	FoundationPose	all_motions	1.1	-0.84	0.36	-0.61	49.61	35.02	18.77	25.64	229.37	159.18	83.34
obj_000002	base	luxonis	FoundationPose	circ_far	13	-0.83	0.5	12.97	175.82	117.46	62.55	4.55	3.31	1.81	1.91
obj_000002	base	luxonis	FoundationPose	circ_close	19.47	-0.79	0.58	19.44	169.96	115.37	65.49	12.84	3.88	5.12	5.26
obj_000002	base	luxonis	FoundationPose	zoom	17.66	-1.28	-1.1	17.58	1.96	2.08	1.64	8.19	5.01	5.09	2.69
obj_000002	base	luxonis	FoundationPose	all_motions	16.07	-0.94	0.09	16.04	127.42	85.84	47	11.48	228.93	152.82	82.87
obj_000002	base	realsense	FoundationPose	circ_far	1.7	0.34	1.57	0.55	174.85	120.25	59.86	7.24	3.75	1.87	1.82
obj_000002	base	realsense	FoundationPose	circ_close	5.84	1.33	3.24	-4.68	2.82	2.71	2.36	4.33	4.07	3.02	2.85

obj_000002	base	realsense	FoundationPose	zoom	6.01	-0.43	-0.4	5.98	6.63	5.53	3.66	7.46	5.96	5.41	2.81
obj_000002	base	realsense	FoundationPose	all_motions	1.66	0.41	1.51	0.53	80.93	56.14	28.48	11.1	254.56	173.78	85.09
obj_000006	base	luxonis	FoundationPose	circ_far	14.51	-1.68	-0.92	14.39	4.04	6.25	6.87	8.15	8.88	16.33	18.04
obj_000006	base	luxonis	FoundationPose	circ_close	13.44	-0.88	1.83	13.29	4.41	6.98	7.32	8.08	10.77	15.09	16.17
obj_000006	base	luxonis	FoundationPose	zoom	22.37	-2.88	-3.83	21.85	6.22	26.6	27.25	10.71	6.6	24.74	23.75
obj_000006	base	luxonis	FoundationPose	all_motions	16.21	-1.78	-0.92	16.08	4.73	11.93	12.48	13.59	9.31	32.58	33.07
obj_000006	base	realsense	FoundationPose	circ_far	1.02	-0.85	-0.57	-0.04	3.48	4.75	3.8	8.69	4.77	9.56	10.26
obj_000006	base	realsense	FoundationPose	circ_close	1.97	-0.24	0.87	1.75	3.37	4.75	4.39	6.46	5.71	11.4	12.94
obj_000006	base	realsense	FoundationPose	zoom	8.98	-2.01	-5.34	6.93	6.29	62.83	63.18	7.45	2.22	8.61	8.57
obj_000006	base	realsense	FoundationPose	all_motions	2.92	-0.99	-1.45	2.34	4.21	20.37	19.94	12.26	5.92	78.05	79.59
obj_000021	base	luxonis	FoundationPose	circ_far	11.38	-0.16	1.83	11.24	1.52	2.07	1.72	8.34	1.65	2.09	2.61
obj_000021	base	luxonis	FoundationPose	circ_close	11.07	0.83	4.03	10.28	2.03	2.15	1.97	5.87	2.25	1.68	2.85
obj_000021	base	luxonis	FoundationPose	zoom	16.46	-0.92	1.06	16.4	0.67	0.66	0.72	10.77	0.91	0.83	1.19
obj_000021	base	luxonis	FoundationPose	all_motions	12.56	-0.08	2.25	12.36	1.43	1.71	1.52	11.55	2.28	2.58	2.82
obj_000021	base	realsense	FoundationPose	circ_far	2.73	0.44	1.7	-2.1	1.25	1.47	1.66	5.99	1.5	2.96	3.1
obj_000021	base	realsense	FoundationPose	circ_close	3.71	1.6	2.94	-1.58	1.07	0.96	1.31	4.45	1.77	1.81	1.79
obj_000021	base	realsense	FoundationPose	zoom	3.99	-1.2	0.08	3.8	1.27	0.93	1.44	6.25	1.86	1.05	1.97
obj_000021	base	realsense	FoundationPose	all_motions	1.69	0.33	1.62	-0.36	1.2	1.18	1.5	9.02	1.69	2.39	2.53

A.3.6. Versuch C3

object_name	occlusion	sensor	method	motion	AP_p	ap_x	ap_y	ap_z	ap_a	ap_b	ap_c	RP_i	RP_a	RP_b	RP_c
obj_000014	base	luxonis	FoundationPose	circ_far	5.41	-1.43	1.03	5.12	1.8	1.91	1.52	5.13	2.6	3.7	2.49
obj_000014	base	luxonis	FoundationPose	circ_close	886.73	-60.82	48.14	883.33	90.84	51.2	105.9	720.66	104.9	46.95	138.67
obj_000014	base	luxonis	FoundationPose	zoom	23.84	1.33	-4.34	23.41	2.52	2.87	3.17	148.96	8.84	16.53	15.62
obj_000014	base	luxonis	FoundationPose	all_motions	260.66	-17.57	12.98	259.74	27.31	16.18	31.64	1046.29	132.65	71.4	158.96
obj_000014	base	realsense	FoundationPose	circ_far	6.27	-1.04	1.26	-6.06	1.21	1.36	1.15	5.75	1.08	1.71	2.18
obj_000014	base	realsense	FoundationPose	circ_close	278.11	77.84	-99.35	247.82	97.06	40.94	109.64	185.39	134.63	88.04	90.77
obj_000014	base	realsense	FoundationPose	zoom	3.41	-0.79	-1.92	2.7	174.74	142.92	38.36	40.14	30.22	4.38	10.84
obj_000014	base	realsense	FoundationPose	all_motions	76.65	21.33	-28.04	68.08	74.88	50.54	41.81	328.95	229.54	181.26	144.22
obj_000002	base	luxonis	FoundationPose	circ_far	12.23	-0.01	1.28	12.16	175.81	117.5	62.5	3.98	3.84	1.73	1.71
obj_000002	base	luxonis	FoundationPose	circ_close	131.59	-71.2	43.69	101.67	63.6	68.83	84.35	90.15	56.73	49.99	41.51
obj_000002	base	luxonis	FoundationPose	zoom	18.29	-0.39	0.13	18.29	2.24	2.3	1.82	7.63	4.64	4.9	2.69
obj_000002	base	luxonis	FoundationPose	all_motions	46.05	-20.32	13.01	39.23	97.17	72.63	52.35	154.77	224.95	144.48	98.64
obj_000002	base	realsense	FoundationPose	circ_far	2.7	0.81	2.56	-0.21	174.35	120.16	59.9	4.16	4.83	1.39	1.5
obj_000002	base	realsense	FoundationPose	circ_close	6.84	1.99	4.18	-5.03	169.48	121.29	59.58	9.22	2.48	4.62	4.5
obj_000002	base	realsense	FoundationPose	zoom	2.79	0.43	0.92	2.6	7.52	6.34	4.09	6.48	6.66	5.95	3.09
obj_000002	base	realsense	FoundationPose	all_motions	2.9	1.04	2.58	-0.82	128.13	89.89	44.81	9.62	219.69	152.15	74.19
obj_000006	base	luxonis	FoundationPose	circ_far	14.39	-1.08	-1	14.32	4.24	7	7.16	8.89	8.24	21.13	22.44
obj_000006	base	luxonis	FoundationPose	circ_close	13.4	-0.57	1.74	13.27	3.85	5.46	5.56	7.2	6.93	12.35	12.75
obj_000006	base	luxonis	FoundationPose	zoom	23.2	-2.4	-3.52	22.8	6.26	27.24	27.74	11.41	5.77	25.22	24.33
obj_000006	base	luxonis	FoundationPose	all_motions	16.38	-1.29	-0.9	16.31	4.67	12.02	12.25	14.23	7.83	34.45	35.08
obj_000006	base	realsense	FoundationPose	circ_far	0.94	-0.69	-0.58	0.26	3.53	4.65	3.5	6.68	4.44	8.63	9.54
obj_000006	base	realsense	FoundationPose	circ_close	1.86	0.03	0.99	1.58	3.52	4.83	4.09	6.01	4.77	10.44	11.59
obj_000006	base	realsense	FoundationPose	zoom	7.12	-1.65	-5.35	4.4	5.83	63.62	63.91	6.69	2.31	10.9	10.89
obj_000006	base	realsense	FoundationPose	all_motions	2.37	-0.74	-1.42	1.75	4.15	20.55	19.9	10.03	5.1	79.01	80.8
obj_000021	base	luxonis	FoundationPose	circ_far	10.98	-0.14	2.33	10.73	1.32	1.88	1.68	7.88	1.71	2.18	2.6
obj_000021	base	luxonis	FoundationPose	circ_close	10.94	0.83	4.59	9.9	1.78	1.9	2	5.66	2.31	1.87	2.77
obj_000021	base	luxonis	FoundationPose	zoom	16.61	-1.32	1.84	16.45	0.74	0.81	0.8	10.21	0.81	1.13	1.33
obj_000021	base	luxonis	FoundationPose	all_motions	12.36	-0.18	2.84	12.03	1.3	1.6	1.53	11.71	2.08	2.35	2.76
obj_000021	base	realsense	FoundationPose	circ_far	3.22	-0.1	2.2	-2.35	1.33	1.61	1.65	5.4	1.54	3.11	3.21
obj_000021	base	realsense	FoundationPose	circ_close	4	1.15	3.37	-1.83	1.15	1.07	1.32	4.92	1.72	1.79	1.85
obj_000021	base	realsense	FoundationPose	zoom	2.25	-1.69	0.74	1.29	1.09	0.84	1.22	5.23	1.48	0.87	1.75
obj_000021	base	realsense	FoundationPose	all_motions	2.47	-0.18	2.14	-1.23	1.21	1.25	1.44	7.38	1.61	2.54	2.6