

## Marius Moosmann

»Autonome Verhakungserkennung und  
Enthakung beim Griff-in-die-Kiste mit  
einem Industrieroboter«





# STUTTGARTER BEITRÄGE ZUR PRODUKTIONSFORSCHUNG BAND 165

**Marius Moosmann**

»Autonome Verhakungserkennung und Enthakung beim Griff-in-die-Kiste  
mit einem Industrieroboter«

## **Herausgeber**

Univ.-Prof. Dr.-Ing. Thomas Bauernhansl<sup>1,2</sup>

Univ.-Prof. Dr.-Ing. Dipl.-Kfm. Alexander Sauer<sup>1,3</sup>

Univ.-Prof. Dr.-Ing. Kai Peter Birke<sup>4</sup>

Univ.-Prof. Dr.-Ing. Marco Huber<sup>1,2</sup>

<sup>1</sup> Fraunhofer-Institut für Produktionstechnik und Automatisierung IPA, Stuttgart

<sup>2</sup> Institut für Industrielle Fertigung und Fabrikbetrieb (IFF) der Universität Stuttgart

<sup>3</sup> Institut für Energieeffizienz in der Produktion (EEP) der Universität Stuttgart

<sup>4</sup> Institut für Photovoltaik (ipv) der Universität Stuttgart

**Kontaktadresse:**

Fraunhofer-Institut für Produktionstechnik und Automatisierung IPA  
Nobelstr. 12  
70569 Stuttgart  
Telefon 0711 970-1100  
info@ipa.fraunhofer.de  
www.ipa.fraunhofer.de

**Bibliographische Information der Deutschen Nationalbibliothek**

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliographie; detaillierte bibliografische Daten sind im Internet über <http://dnb.de> abrufbar.

Zugl.: Stuttgart, Univ., Diss., 2023

**D 93**

2024

**Druck und Weiterverarbeitung:**

Fraunhofer-Druckerei, Stuttgart  
Für den Druck des Buches wurde chlor- und säurefreies Papier verwendet.



Dieses Werk steht, soweit nicht gesondert gekennzeichnet, unter folgender Creative-Commons-Lizenz:  
Namensnennung – Nicht kommerziell – Keine Bearbeitungen  
International 4.0 (CC BY-NC-ND 4.0).

# **Autonome Verhakungserkennung und Enthakung beim Griff-in-die-Kiste mit einem Industrieroboter**

**Von der Fakultät Konstruktions-, Produktions- und  
Fahrzeugtechnik der Universität Stuttgart  
zur Erlangung der Würde eines Doktor-Ingenieurs (Dr.-Ing.)  
genehmigte Abhandlung**

**Vorgelegt von**

**Marius Moosmann**

**aus Calw**

Hauptberichter: Univ.-Prof. Dr.-Ing. habil. Marco F. Huber

Mitberichter: Hon.-Prof. Dr.-Ing. habil. Andreas Pott

Tag der mündlichen Prüfung: 30. Oktober 2023

Institut für Industrielle Fertigung und Fabrikbetrieb der Universität Stuttgart

2023



# Danksagung

Die vorliegende Arbeit entstand während meiner Tätigkeit als wissenschaftlicher Mitarbeiter am Institut für Produktionstechnik und Automatisierung (IPA) der Fraunhofer Gesellschaft in Stuttgart.

Mein großer Dank gilt meinem Doktorvater Herrn Prof. Dr.-Ing. Marco F. Huber für die Förderung und Unterstützung meiner wissenschaftlichen Tätigkeit und dieser Arbeit. Bei Herrn Prof. Dr.-Ing. Andreas Pott bedanke ich mich für die wertvollen Diskussionen und Beiträge und die Übernahme des Mitberichts.

Ich danke allen meinen Kollegen für die wertvolle Zusammenarbeit und inhaltlichen Diskussionen. Besonderer Dank gilt Dr.-Ing. Felix Spenrath und Dipl.-Ing. Richard Bormann, M.Sc. für die Durchsicht meiner Arbeit und die angenehme Arbeitsatmosphäre. Für die Unterstützung meines Promotionsvorhabens möchte ich mich zudem bei Herrn Manuel Mönnig, M.Sc. bedanken. Zum Gelingen der Arbeit haben meine Studierenden beigetragen, welchen ich ebenfalls danken möchte. Bei Frau Heide Kreuzburg möchte ich mich für die organisatorische Unterstützung des Promotionsprozesses bedanken.

Mein herzlicher Dank gilt meiner Familie, die mich stets bedingungslos und uneingeschränkt unterstützt und gefördert hat.

*Ich widme diese Arbeit meiner Schwester Aline und meinen Eltern Susanne und Klaus.*





# Kurzfassung

In dieser Arbeit wird eine Methode zur Verhakungserkennung und Enthakung für eine Griff-in-die-Kiste-Applikation vorgestellt. Hierfür werden maschinelle Lernverfahren eingesetzt, welche eine effiziente, skalierbare und flexible Lösung ermöglichen.

Nach der Erfassung des Tiefenbildes von chaotisch gelagerten Werkstücken in einer Kiste, werden deren Lagen durch eine Objektlageerkennung erfasst und die Tiefeninformation in Form eines Tiefenkartenzuschnitts an einen binären Klassifikator übergeben, welcher anhand eines künstlichen neuronalen Netzes die Verhakungswahrscheinlichkeit schätzt. Der Klassifikator wird mit synthetisch generierten Datenbeispielen trainiert, welche vor dem Training durch Transfermethoden an die Realität angenähert werden. Die Prognosen werden als Kosten in einer Heuristik zur Greifplanung berücksichtigt.

Um nach einer Verhakungserkennung verhakte Werkstücksituationen mit einem Industrieroboter auflösen zu können, wird in dieser Arbeit eine Methode zur Enthakung vorgestellt. Über die verhakte Werkstücksituation wird eine Hemisphäre mit möglichen zusätzlichen Pfadposen aufgespannt und diese Posen bewertet und die für eine Enthakung bestmögliche Pose durch zwei überwachte und eine bestärkende Lernmethode prognostiziert. Durch zusätzliche Orientierungsänderungen an den Pfadposen und einem Ablehnungskriterium für unmöglich zu lösende Verhakungen, sowie dem Transfer der in einer Simulation trainierten Methoden in die Realität, werden die ausgewählten Pfadposen kollisionsgeprüft an eine Greifplanung übergeben. Die Greifplanung übergibt die berechneten Pfadposen an eine Robotersteuerung, welche die Entnahme und Enthakung durchführt.

Zum Abschluss wird die in dieser Arbeit entwickelte Methode zur Verhakungserkennung und Enthakung in einer realen Roboterzelle experimentell validiert und gezeigt, dass die Entwicklung zu einer Effizienzsteigerung und gesteigerten Robustheit eines Griff-in-die-Kiste-Gesamtsystems führt.



# Abstract

In this dissertation, a method for entanglement detection and separation for a bin-picking application is presented. Machine learning techniques are used for this purpose, which provide an efficient, scalable and flexible solution.

After acquiring the depth image of chaotically stored workpieces in a bin, their poses are detected by object pose estimation and the depth information is passed in the form of a depth map crop to a binary classifier, which estimates the entanglement probability using an artificial neural network. The classifier is trained with synthetically generated data examples, which are approximated to reality by transfer methods before training. The predictions are considered as costs in a heuristic for path planning.

In order to separate entangled workpieces with an industrial robot after entanglement detection, a method for entanglement separation is presented in this work. A hemisphere with possible additional path poses is spanned over the entangled workpiece situation and these poses are evaluated and the best possible pose for separation is predicted by two supervised and one reinforcement learning methods. Through additional orientation changes to the path poses and a rejection criterion for entanglement that is impossible to separate, as well as the transfer of the methods trained in a simulation to reality, the selected path poses are transferred to a grasp planning in a collision-checked manner. The grasp planning passes the computed path poses to a robot controller, which performs the grasping and separation.

Finally, the method for entanglement detection and separation developed in this work is experimentally validated in a real robot cell and it is shown that the development leads to an increase in efficiency and increased robustness of an bin-picking system.



# Inhaltsverzeichnis

Abkürzungsverzeichnis	xv
Symbolverzeichnis	xvii
Abbildungsverzeichnis	xxi
Tabellenverzeichnis	xxiii
<b>1 Einleitung</b>	<b>1</b>
1.1 Ausgangssituation und Motivation . . . . .	1
1.2 Problemstellung . . . . .	7
1.3 Stand der Forschung . . . . .	10
1.3.1 Modellfreie Ansätze . . . . .	11
1.3.2 Modellbasierte Ansätze . . . . .	14
1.3.3 Schlussfolgerung . . . . .	16
1.4 Zielsetzung und Abgrenzung . . . . .	17
1.5 Lösungsansatz und Gliederung der Arbeit . . . . .	19
<b>2 Grundlagen</b>	<b>23</b>
2.1 Mathematische Grundlagen und Notation . . . . .	23
2.2 Maschinelles Lernen . . . . .	24
2.2.1 Einführung . . . . .	24
2.2.2 Überwachtes Lernen . . . . .	27
2.2.3 Bestärkendes Lernen . . . . .	28
2.2.4 Gefaltete, neuronale Netzwerke . . . . .	30
2.2.5 Statistische Analyse einer binären Klassifizierung . . . . .	31
2.3 Komponenten einer Griff-in-die-Kiste-Vereinzelungslösung . . . . .	33
2.3.1 3D-Bildsensor und Objektlageerkennung . . . . .	33
2.3.2 Industrieroboter . . . . .	36
2.3.3 Greifsysteme . . . . .	38

2.3.4	Greifplanung . . . . .	39
2.3.5	Roboterzellenperipherie . . . . .	43
<b>3</b>	<b>Entwicklung einer Methode zur Verhakungserkennung von Werkstücken</b>	<b>45</b>
3.1	Ausgangslage, Generierung und Aufbereitung benötigter Daten . . . . .	45
3.1.1	Daten der Bildsensorik und der Objektlageerkennung . . . . .	46
3.1.2	Tiefenkartenzuschnitt . . . . .	48
3.1.3	Datengenerierung in einer Simulationsszene . . . . .	50
3.1.4	Augmentierung der Daten . . . . .	56
3.2	Transfer von der Simulation in die reale Roboterzelle . . . . .	56
3.2.1	Domänenrandomisierung . . . . .	57
3.2.2	Inpainting . . . . .	58
3.2.3	Domänenanpassung . . . . .	60
3.3	Gefaltete neuronale Netzwerkarchitektur zur Erkennung von Verhakungen	64
3.4	Einordnung der Verhakungserkennung in das Gesamtsystem . . . . .	66
3.5	Zusammenfassung . . . . .	68
<b>4</b>	<b>Entwicklung eines Verfahrens zur Enthakung von verhakten Werkstücken</b>	<b>69</b>
4.1	Ausgangslage und Aufbereitung vorhandener Daten . . . . .	69
4.1.1	Daten der Bildverarbeitung und Objektlageschätzung . . . . .	70
4.1.2	Daten der Greifplanung . . . . .	70
4.2	Repräsentation von Bewegungspfaden zur Enthakung mit einem Hemisphärenmodell . . . . .	74
4.2.1	Verteilung möglicher Pfadposen auf einer Hemisphäre . . . . .	74
4.2.2	Annotierung der Hemisphärenpfadposen . . . . .	76
4.2.3	Erzeugung von mehreren Hemisphären zur Enthakungspfadplanung	81
4.2.4	Validierung des Hemisphärenmodells von Bewegungspfaden zur Enthakung . . . . .	82
4.2.5	Orientierungsänderung an den Hemisphärenpfadposen . . . . .	90

---

4.3	Überwachte maschinelle Lernmethoden zur Auswahl geeigneter Enthakungspfadposen . . . . .	92
4.3.1	Datengenerierung in einer Simulationsszene . . . . .	92
4.3.2	Enthakungspfadplanung durch eine Reihenschaltung neuronaler Netzwerkarchitekturen . . . . .	95
4.3.3	Enthakungspfadplanung durch eine zusammengeführte, neuronale Gesamtarchitektur . . . . .	97
4.3.4	Netzwerkarchitektur zur Enthakungspfadplanung durch überwachte Lernmethoden . . . . .	98
4.4	Bestärkende maschinelle Lernmethode zur Auswahl geeigneter Enthakungspfadposen . . . . .	101
4.4.1	Training eines Agenten für das bestärkende Lernen . . . . .	101
4.4.2	Netzwerkarchitektur des bestärkenden Lernverfahrens . . . . .	106
4.5	Transfer von der Simulation in die reale Roboterzelle . . . . .	107
4.5.1	Implementierung des Transfers für die überwachten Lernverfahren .	108
4.5.2	Implementierung des Transfers für das bestärkende Lernverfahren .	109
4.6	Einführung eines Ablehnungskriteriums zur Reduzierung situativ unmöglicher Enthakungsstrategien . . . . .	110
4.7	Einordnung der Enthakungspfadplanung in das Gesamtsystem . . . . .	112
4.8	Zusammenfassung . . . . .	113
<b>5</b>	<b>Ergebnisse und Validierung</b>	<b>117</b>
5.1	Beschreibung der Evaluierungsumgebung in der Simulation und in der Realität	117
5.2	Verhakungserkennung im direkten Transfer . . . . .	120
5.3	Vergleich der überwachten Lernverfahren zur Enthakung im direkten Transfer	122
5.4	Vergleich des überwachten und bestärkenden Lernverfahrens zur Enthakung im direkten Transfer . . . . .	123
5.5	Einfluss von Transfermethoden auf die Verhakungserkennung und Enthakung	126
5.5.1	Einfluss verschiedener Randomisierungsmethoden . . . . .	126
5.5.2	Optimierung der Methode zur Verhakungserkennung . . . . .	128
5.5.3	Optimierung der Methode zur Enthakungspfadplanung . . . . .	129

---

5.6	Validierung zusätzlicher Optimierungen der Enthakungsmethodik . . . . .	130
5.6.1	Einfluss zusätzlicher Orientierungsänderungen an den Pfadposen der Hemisphären . . . . .	131
5.6.2	Einfluss eines Ablehnungskriteriums . . . . .	133
5.7	Validierung des Gesamtsystems . . . . .	135
<b>6</b>	<b>Zusammenfassung und Ausblick</b>	<b>141</b>
6.1	Zusammenfassung . . . . .	141
6.2	Ausblick . . . . .	142
	<b>Literatur</b>	<b>145</b>
<b>A</b>	<b>Anhang</b>	<b>163</b>
A.1	Winkel der Rotationsmodelle in Abhängigkeit der Hemisphärenpfadpose .	163



# Abkürzungsverzeichnis

<b>CAD</b>	Computer Aided Design
<b>CNN</b>	Konvolutionales, neuronales Netzwerk
<b>FN</b>	False Negative
<b>FP</b>	False Positive
<b>GPU</b>	Graphical Processing Unit
<b>KMS</b>	Kraft-Momenten-Sensor
<b>Laser</b>	Light amplification by stimulated emission of radiation
<b>LED</b>	Lichtemittierende Diode
<b>LIN</b>	Linearbewegung
<b>MLP</b>	Mehrschichtperzeptron
<b>PTP</b>	Point-to-Point-Bewegung
<b>ReLU</b>	Rectified Linear Unit
<b>STL</b>	Standard Triangulation Language
<b>TCP</b>	Tool Center Point / Tool Center Pose
<b>TN</b>	True Negative
<b>TP</b>	True Positive
<b>VRML</b>	Virtual Reality Modeling Language



# Symbolverzeichnis

## Griechische Symbole

Symbol	Einheit	Beschreibung
$\alpha$	-	Lernrate
$\alpha_P$	$^\circ$	Aufteilungswinkel Pfadposen auf Hemisphäre
$\gamma$	-	Diskontierungsfaktor
$\epsilon$	-	Zufallsfaktor der Aktionsauswahl
$\theta$	$^\circ$	Winkel der Drehzeigerdarstellung
$\pi$	-	Strategie
$\pi^*$	-	optimale Strategie
$\sigma_G$	-	Varianz Gauß'sches Rauschen
$\sigma_H$	-	Varianz Helligkeit
$\sigma_R$	$^\circ$	Varianz Rotation
$\sigma_T$	Millimeter	Varianz Translation
$\phi$	-	Drehwinkelschrittweite
$\omega$	-	Gewichtungsfaktor

## Lateinische Symbole

Symbol	Einheit	Beschreibung
$A$	-	endliche Menge an Aktionen
$D_X$	-	Diskriminator von Simulation zu Realität
$D_Y$	-	Diskriminator von Realität zu Simulation
$G_X$	-	Generator von Simulation zu Realität
$G_Y$	-	Generator von Realität zu Simulation
$K_0$	-	Welt-Koordinatensystem
$K_L$	-	Koordinatensystem des Ladungsträgers

<b>Symbol</b>	<b>Einheit</b>	<b>Beschreibung</b>
$K_W$	-	Koordinatensystem des Werkstücks
$K$	-	Koordinatensystem
$L_{Ent}$	-	Verlustfunktion der Enthakungspfadplanung
$L_{G_x}$	-	Kreuzentropie-Verlustfunktion Generator Simulation zu Realität
$L_{G_y}$	-	Kreuzentropie-Verlustfunktion Generator Realität zu Simulation
$L_{VE}$	-	Verhakungserkennungsverlust
$L_{CE}$	-	Verlustfunktion der Kreuzentropie
$L_{MSE}$	-	Verlustfunktion der mittleren quadratischen Abweichung
$L_{cyc}$	-	Zykluskonsistenzverlust
$L_{total}$	-	Gesamtverlust
$N$	-	Anzahl vorhandener Datensätze
$Q(s,a)$	-	Zustand-Aktions-Paar
$R(s)$	-	Belohnungsfunktion
$S$	-	endliche Menge an Zuständen
$a_i$	-	Ausgangswert eines künstlichen Neurons
$a$	-	Aktion
$b$	-	Schrittweite
$c_O$	-	Objektlageerkennungskostenfaktor
$c_U$	-	Additionsfaktor Belohnung
$c_V$	-	Verhakungsverlustfaktor
$c_d$	-	dynamischer Kostenfaktor
$c_s$	-	statischer Kostenfaktor
$c_{n_P}$	-	Gesamtkostenbewertung der Hemisphärenpfadpose
$d_E$	Millimeter	Abstand der Hemisphärenebenen
$d$	-	Bewertungsfaktor
$g_i$	-	Gewichtungsfaktor eines künstlichen Neurons
$h(x)$	-	Heuristikfunktion
$h_E$	Millimeter	Höhe der Hemisphärenebene

<b>Symbol</b>	<b>Einheit</b>	<b>Beschreibung</b>
$h_G$	Millimeter	Grenzwert der Höhenausdehnung der Kiste
$h_K$	Millimeter	Höhe des Kistenrandes
$i$	-	Laufvariable
$j$	-	Laufvariable
$k$	-	Laufvariable
$l$	-	Grundwahrheit
$m$	-	Laufvariable
$n_E$	-	Anzahl der Hemisphärenebenen
$n_H$	-	Anzahl an Hemisphären
$n_P$	-	Anzahl an Hemisphärenpfadposen
$n_V$	-	Anzahl an Verhakungen
$n_f$	-	Interpolationsschritte der Translation
$n_\varphi$	-	Interpolationsschritte der Rotation
$n$	-	Laufvariable
$p_g$	-	Grenzwert des Ablehnungskriteriums
$p_{max}$	-	Maximalwert der Prognosen
$p$	-	Pose
$r_H$	Millimeter	Radius der Hemisphäre
$s$	-	Zustand
$t$	-	Zeitfaktor
$w_P$	-	Gewichtungsfaktor der Pfadpose
$w_V$	-	Gewichtungsfaktor Verhakungsverlustfaktor
$w_d$	-	Gewichtungsfaktor Heuristikkostenfaktoren
$x(t)$	-	Eingangswert
$x$	-	Eingangswert
$y_i$	-	Soll-Wert
$y$	-	Wert auf der y-Achse
$\hat{y}_i$	-	prognostizierter Wert
$z$	-	Wert auf der z-Achse

## Vektoren und Matrizen

Symbol	Einheit	Beschreibung
$I$	-	Bildmatrix
$M_a$	-	Matrix des Hemisphärenindex
$P_{s,s'}^a$	-	Transitionsmatrix
$R$	-	Rotationsmatrix
$T$	-	Homogene Transformationsmatrix
$\vec{\omega}$	-	Richtung des Drehzeigers
$\vec{p}$	-	Ortsvektor

## Indizes

Symbol		Beschreibung
0	-	Bezug zum Ursprung (Welt)
$F$	-	Bezug zum Roboterflansch
$H$	-	Bezug zur Hemisphäre
$L$	-	Bezug zum Ladungsträger
$T$	-	Bezug zum Werkzeugmittelpunkt
$W$	-	Bezug zum Werkstück
$\varphi$	-	Bezug zum rotatorischen Freiheitsgrad
$f$	-	Bezug zum translatorischen Freiheitsgrad

# Abbildungsverzeichnis

1.1	Aufbau und Ablauf einer Griff-in-die-Kiste-Applikation . . . . .	4
1.2	Erfolgsquoten im Dauertest von zwei Roboteranlagen . . . . .	6
1.3	Fehleranalyse im Dauertest von zwei Roboteranlagen . . . . .	7
1.4	Verhakte Werkstücksituation von Pleuelstangen in der Kiste . . . . .	9
2.1	3D-Bildsensoren . . . . .	34
2.2	Industrieroboter . . . . .	37
2.3	Entnahmepfadposen am Beispiel eines Klemmgreifers . . . . .	42
3.1	Punktwolke einer realen Kistensituation . . . . .	46
3.2	Koordinatensysteme und Transformationsmatrizen . . . . .	47
3.3	Beispiel eines Tiefenkartenzuschnitts . . . . .	48
3.4	Zerlegung der konkaven Geometrie in mehrere konvexe Körper . . . . .	51
3.5	Simulationsablauf zur Datengenerierung für die Verhakungserkennung . . . . .	52
3.6	Simulationsszene in der Software CoppeliaSim . . . . .	53
3.7	Unmögliche Verhakungssituation in der Kiste . . . . .	55
3.8	Tiefenkartenzuschnitte von Pleuelstangen aus der Simulation . . . . .	55
3.9	Horizontale Spiegelung der Tiefenkarte . . . . .	56
3.10	Transformation einer synthetischen, originalen Tiefenkarte . . . . .	59
3.11	Aufbau und Ablauf einer Domänenanpassung . . . . .	61
3.12	<i>CycleGAN</i> Generator Netzwerkarchitektur . . . . .	62
3.13	<i>CycleGAN</i> Diskriminator Netzwerkarchitektur . . . . .	63
3.14	Netzwerkarchitektur der Verhakungserkennung . . . . .	64
3.15	Ablaufschema und Interaktion einer Verhakungserkennung . . . . .	66
4.1	Übersicht der zu vergleichenden Enthakungsmethoden . . . . .	70
4.2	Übersicht von Koordinatensystemen und Transformationen . . . . .	71
4.3	Rotatorische und translatorische Freiheitsgrade von definierten Greifposen . . . . .	72

---

4.4	Hemisphärenlage relativ zu einem lokalisierten Werkstück . . . . .	75
4.5	Verteilung der Hemisphärenpfadposen auf den Hemisphären . . . . .	76
4.6	Enthakungsstrategien . . . . .	82
4.7	Simulationsabbildungen der Werkstückgeometrien . . . . .	83
4.8	Bestimmung des Hemisphärenradius . . . . .	86
4.9	Dreidimensionale Ansicht einer Enthakungstrajektorie . . . . .	88
4.10	Bestimmung des Gewichtungsfaktors $w_P$ . . . . .	89
4.11	Rotationsmodell (a) und (b) . . . . .	90
4.12	Simulationsablauf zur Datengenerierung für die Enthakungsmethode . . . . .	93
4.13	Reihenschaltung von neuronalen Netzwerken zur Enthakungspfadplanung . . . . .	96
4.14	Zusammengeführte Netzwerkarchitektur zur Enthakungspfadplanung . . . . .	98
4.15	Netzwerkarchitektur der überwachten Lernmethoden . . . . .	100
4.16	Trainingsumgebung, Bestandteile und Interaktion des Agenten . . . . .	102
4.17	Netzwerkarchitektur des bestärkenden Lernverfahrens . . . . .	108
4.18	Trainingsschema des bestärkenden DQN-Agententrainings . . . . .	109
4.19	Nicht enthakbare Werkstücksituation von Pleuel . . . . .	110
4.20	Bestimmung des Grenzwertes zur Ablehnung von unmöglicher Enthakung . . . . .	112
4.21	Ablaufschema und Interaktion einer Enthakungsplanung . . . . .	113
4.22	Ablauf einer Enthakung von Rundstahlbügeln . . . . .	114
4.23	Ablauf einer Enthakung von Pleueln . . . . .	114
5.1	Versuchsaufbau einer realen Griff-in-die-Kiste-Roboterzelle . . . . .	118
5.2	Verhakte Werkstücksituationen in einer Kiste . . . . .	119
5.3	Histogramm der Berechnungszeit der Verhakungserkennung und Enthakung . . . . .	138



# Tabellenverzeichnis

4.1	Validierung der Anzahl der Hemisphärenpfadposen . . . . .	84
4.2	Anzahl an Bewegungen und benötigte Zeit in der Simulation . . . . .	85
4.3	Vergleich der Strategien zur Enthakung . . . . .	87
5.1	Analyse der Methode zur Verhakungserkennung im direkten Transfer . . . . .	121
5.2	Vergleich der überwachten Lernverfahren im direkten Transfer . . . . .	123
5.3	Vergleich der benötigten Daten und der Dauer der Datengenerierung . . . . .	123
5.4	Vergleich der Lernverfahren zur Enthakung im direkten Transfer . . . . .	124
5.5	Vergleich der Daten, Dauer der Datengenerierung und Gesamtdauer . . . . .	125
5.6	Untersuchung der Eignung verschiedener Randomisierungsmethoden . . . . .	127
5.7	Vergleich direkter Transfers mit Domänenrandomisierung und -anpassung . . .	128
5.8	Vergleich der Transfermethoden für die Enthakungspfadplanung . . . . .	130
5.9	Validierung des Einflusses zusätzlicher Orientierungsänderungen . . . . .	131
5.10	Enthakungsversuche mit zusätzlichen Orientierungsänderungen . . . . .	132
5.11	Validierung des Einflusses eines zusätzlichen Ablehnungskriteriums . . . . .	134
5.12	t-Test zur Ermittlung der statistischen Signifikanz . . . . .	134
5.13	Validierung des Griff-in-die-Kiste-Gesamtsystems . . . . .	136
5.14	Validierung des Griff-in-die-Kiste-Gesamtsystems . . . . .	137
A.1	Winkel des Rotationsmodells (a) . . . . .	163
A.2	Winkel des Rotationsmodells (b) . . . . .	164



# 1 Einleitung

Dieses Kapitel stellt die Thematik der automatisierten Werkstückvereinzelung und deren Herausforderung mit sich verhakenden Werkstückgeometrien vor. Zu Beginn wird hierzu die Ausgangssituation erläutert und die Motivation dieser Arbeit aufgezeigt. Ferner wird die Problemstellung der sich verhakenden Werkstücke detailliert und es werden bisherige Arbeiten zur Lösung dieser Problematik im Stand der Forschung genannt und beschrieben. In Abschnitt 1.4 wird diese Arbeit thematisch abgegrenzt und die Zielsetzung mit der Nennung der Forschungsfrage aufgezeigt. Am Ende des Kapitels wird der beschriebene Lösungsansatz kurz aufgezeigt und die daraus resultierende Gliederung der Arbeit vorgestellt.

## 1.1 Ausgangssituation und Motivation

Produzierende Unternehmen haben in den vergangenen Jahren erkannt, dass Produkte, welche auf den individuellen Geschmack und Nutzen des Verbrauchers abgestimmt sind, immer mehr Zuspruch und Nachfrage erlangen. Dies wird im englischen als *Low Volume High Mix* oder *Mass Personalization* bezeichnet. Während Henry Ford noch marktdeckend mit einer Produktvariante fuhr und diese mit seinem weltbekannten Spruch vermarktete, „*Any customer can have a car painted any color that he wants so long as it is black.*“ (Ford & Crowther 1922), herrscht im heutigen Zeitalter ein Paradigmenwechsel und ein Aufbruch in den Produktindividualismus. Produkte werden vermehrt auf den Kunden zugeschnitten, was eine deutlich höhere Vernetzung und komplexere Steuerung des Produktionsprozesses verlangt. Individualisierte Produkte bedeuten im Umkehrschluss eine hohe Varianz der Produktbestandteile und damit insgesamt sinkende Losgrößen der Werkstücktypen, welche für Produktionsprozesse automatisiert gehandhabt werden müssen. Dies stellt für Vereinzelungsprozesse allerdings eine große Schwierigkeit dar.

Als automatisierte Handhabungssysteme von Produkten oder Produktbestandteilen werden Anlagen, Maschinen oder Robotersysteme bezeichnet, die den Materialfluss von einer Prozesseinheit zu einer anderen vollziehen können. Hiermit ist außerdem die Vereinzelung aus einem Ladungsträger und die Ablage des Werkstücks in einer Vorrichtung oder Maschine zur Weiterverarbeitung gemeint.

Viele Werkstücke werden chaotisch in Behältern oder Kisten zu produzierenden Unternehmen und an die jeweiligen Prozesse geliefert. Einer der Gründe hierfür ist die kostengünstige Lagerung. Unternehmen müssten einen zusätzlichen Prozess einsetzen, um die Werkstücke nach dem Produktionsprozess in eine geordnete Lage zu bringen. Dieser kann durch eine Arbeitskraft oder durch ein zusätzliches Robotersystem mit Bildverarbeitung erfolgen. Ferner stellen die, wie bereits beschrieben, immer kleiner werdenden Losgrößen hierfür ein Problem dar. Um die geordnete Lage der Werkstücke auch während des Transports zu gewährleisten, sind meist Sonderladungsträger und Blister notwendig. Diese müssen aufwändig geplant und gefertigt werden, was bei kleineren Losgrößen nicht wirtschaftlich ist.

Die automatisierte Vereinzelung von Werkstücken ist seit mehr als 35 Jahren Bestandteil der angewandten Forschung (Ikeuchi et al. 1983; Horn & Ikeuchi 1983; Horn & Ikeuchi 1984) und erfährt aufgrund der bereits erläuterten Thematik des *Low Volume High Mix* und der *Mass Personalization* stets an Bedeutung. Gerade auch für Hochlohnländer wie die Bundesrepublik Deutschland stellen automatisierte Vereinzelungstechnologien eine wirtschaftliche Alternative zur manuellen Vereinzelung von Werkstücken im Produktionsprozess dar.

Für die automatisierte Werkstückvereinzelung werden unterschiedliche Methoden eingesetzt. Eine der meistgenutzten Methoden ist der Einsatz von Wendelförderanlagen. Anhand von mechanischen Schikanen werden Werkstücke als Schüttgut in einen rotatorisch vibrierenden Topf gefördert und hier ausgerichtet. Falsch orientierte Werkstücke fallen dabei im Verlauf der Förderstrecke wieder zurück in den Topf. Nur richtig ausgerichtete Werkstücke gelangen bis an das Ende der Förderstrecke und können danach vereinzelt werden. Diese Vereinzelungsmethode wird vermehrt bei kleineren Werkstückabmessungen und -gewichten

eingesetzt, da große und auch schwere Werkstücke meist mit erheblichem Lärm gefördert werden. Ein weiteres Problem dieser Methode stellt der potenzielle Verschleiß der Bauteile und der Maschine dar. Gerade bei sensiblen Werkstückoberflächen oder -geometrien, wie beispielsweise Gewindegängen, kann diese Methode zu Beschädigungen und damit zu Ausschuss führen. Die Gestaltung der Schikanen ist zudem meist sehr aufwändig, weshalb es sich für kleine Losgrößen eher als ungeeignet darstellt.

Um Prozesse schnell auf wechselnde Werkstücke adaptieren zu können, findet die Bildgebung mittels Kamerasystem und verbundener Bildverarbeitung mehr und mehr Einzug in die automatisierte Werkstückvereinzelung. Hierbei unterscheidet man zwischen zweidimensionaler und dreidimensionaler Bildverarbeitung.

Um zweidimensionale Bildverarbeitung einsetzen zu können, werden die Werkstücke vor der Vereinzelung beispielsweise auf einer Fläche oder einem Förderband ausgeschüttet, welches die Werkstücke aus einer dreidimensionalen Lage in der Kiste auf eine zweidimensionale Lage auf einer flachen Ebene vorbereitet. Mittels zweidimensionaler Kameratechnik und Algorithmik können die Werkstücke abgegriffen werden. Hierfür kommen günstigere Komponenten zum Einsatz, wie es im Gegensatz für eine Applikation mit dreidimensionaler Bildverarbeitung benötigt wird. Dennoch ist die Vorvereinzelung auf einer flachen Ebene nicht für alle Werkstückgeometrien geeignet. Diese Methode bietet sich insbesondere für flache, kleine und leichte Objekte an, welche schnell gehandhabt werden müssen. Ein Beispiel hierfür sind klassische Pick-and-Place Anwendungen in der Lebensmittelindustrie oder auch die Vereinzelung von Schrauben. Für große, schwere und auch leicht zu beschädigende Werkstücke ist diese Methode nicht geeignet. Auch hier kann es zu Lärmbelästigungen und Beschädigungen kommen. Die Vorbereitung der Werkstücke für zweidimensionale Bildverarbeitungssysteme benötigt zudem mehr Hardware und damit Platzbedarf, beispielsweise für die Installation von Förderbändern und Auskipmechaniken.

Schwere Werkstückgeometrien stellen bei manueller Entnahme Gesundheitsrisiken für einen Mitarbeiter dar, da große Werkstücke meist in großen Kisten gelagert werden und dadurch unergonomische Bewegungen durchgeführt werden müssen. Für diese Szenarien stellt die

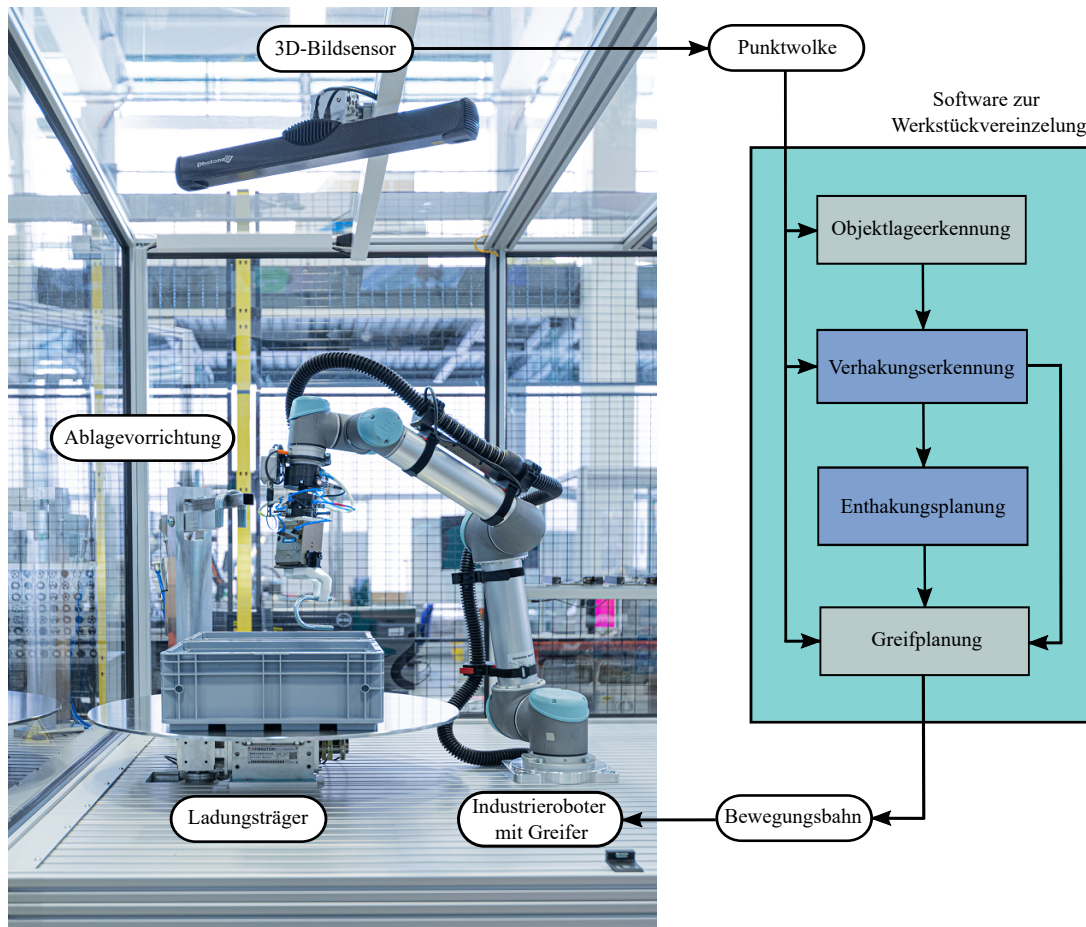


Abbildung 1.1: Aufbau und Ablauf einer Griff-in-die-Kiste-Applikation (Quelle des Originalfotos: Fraunhofer IPA)

automatisierte Vereinzelnung mit einem Griff in diese Kisten durch einen Industrieroboter in Verbindung mit einem dreidimensionalen Kamerasystem eine weitere Methode dar. Diese Methode wird meist *Griff-in-die-Kiste* oder (*Random*) *Bin Picking* genannt. Die Kosten für eine dreidimensionale Bildsensorik sowie für einen Manipulator, welcher hierfür benötigt wird, stellen allerdings die Wirtschaftlichkeit eines solchen Systems in Frage. Zudem handelt es sich um die komplexeste der vorgestellten Methoden, weshalb meist die Robustheit solcher Anwendungen angezweifelt wird.

Ein Entnahmeprozess beim Griff-in-die-Kiste beginnt, wie in Abbildung 1.1 dargestellt, mit der Aufnahme einer Punktwolke durch einen 3D-Bildsensor, welche das Tiefenbild einer unter dem Sensor befindlichen Kiste mit Werkstücken repräsentiert. Diese Punktwolke dient als Eingangsinformation für eine Software zur Werkstückvereinzelnung. Die Software besteht

aus einem Modul zur Objektlageerkennung, einem Greifplanungsmodul und potentiell zusätzlichen Modulen, wie beispielsweise einer Verhakungserkennung und Enthakungsplanung, welche in dieser Arbeit entwickelt werden. Die Objektlageerkennung lokalisiert vordefinierte Werkstückgeometrien in der gegebenen Kistensituation und übergibt die Ergebnisse an die folgenden Module. Die Greifplanung sucht neben möglichen Greifposen auch Entnahmepfade, welche kollisionsfrei durch einen Industrieroboter abgefahren werden können. Diese gefundenen Pfadposen werden dann an eine Anlagen- oder Robotersteuerung übergeben, welche den Industrieroboter zur gewünschten Greifpose leitet (Spenrath et al. 2013). Nachdem eine erfolgreiche Entnahme eines Werkstücks durchgeführt wurde, kann das gegriffene Werkstück in eine Ablagevorrichtung eingeführt oder abgelegt werden. Der Prozess startet daraufhin erneut mit einer weiteren Bildaufnahme.

Industrieroboter werden sehr häufig bei repetitiven Aufgaben, wie beispielsweise dem Schweißen und Lackieren, eingesetzt. Hierfür bedarf es einer einmaligen Programmierung durch einen Mitarbeiter, sodass verschiedene vordefinierte Programme abgerufen werden können. Meist sind hierbei Roboterposen fix definiert und werden je nach Prozess abgefahren. Bei der Vereinzelung von Schüttgut hingegen werden Roboterposen in Abhängigkeit der zufälligen Werkstücklage in der Kiste berechnet und ausgeführt.

Laut dem World Robotics Report (2022) waren im Jahr 2022 3.477.000 Industrieroboter im operativen Einsatz. Handhabungsaufgaben werden von ungefähr 46% der Industrieroboter durchgeführt, worunter auch Vereinzelungen mittels Griff-in-die-Kiste-Applikationen fallen. Dennoch ist der Anteil der Griff-in-die-Kiste-Applikationen verhältnismäßig gering. Hierfür gibt es mehrere Gründe und es ist notwendig, den Prozess des Griff-in-die-Kiste näher zu beleuchten und Anforderungen an diesen Prozess zu deklarieren.

Neben wirtschaftlichen Faktoren wie der Größe der Roboteranlage sowie deren Kosten und Instandhaltung, werden technische Faktoren wie Prozess- und Rüstzeiten betrachtet. Automatisierte Vereinzelungsprozesse werden sehr oft mit der menschlichen Leistung verglichen, Werkstücke zu vereinzeln und diese lagerichtig abzulegen. Durch die enorme Taktilität einer menschlichen Hand ist ein Mitarbeiter schneller und auch robuster in der Durch-

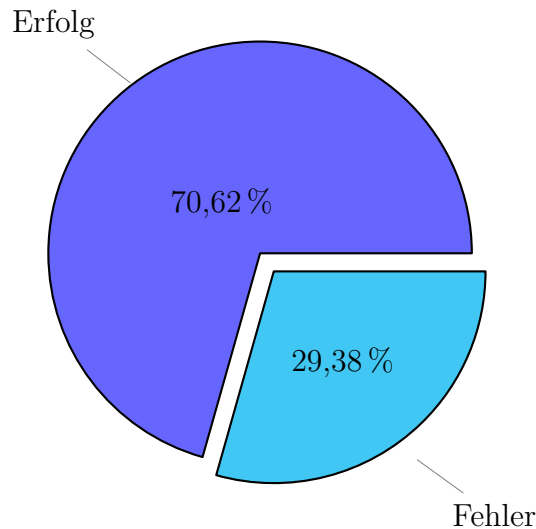


Abbildung 1.2: Erfolgsquoten im Dauertest von zwei Roboteranlagen zur Vereinzelung von Pleuelstangen nach (Spenrath 2022)

führung einer Werkstückentnahme aus einer Kiste. In zweidimensionalen *Pick-and-Place* Anwendungen sind eingesetzte Industrieroboter meist schneller als der Mensch. Ein Grund hierfür ist die zweidimensionale Lage der Werkstücke. Zudem muss keine Kiste erkannt und betrachtet werden, die für eine Maschine eine mögliche Kollisionsgefahr darstellt. Ferner sind dreidimensionale im Vergleich zu zweidimensionalen Bildverarbeitungsalgorithmen aufgrund der erhöhten Datenmenge üblicherweise deutlich rechenintensiver und langsamer. Meist lohnen sich vollautomatisierte Prozesse nur dann, wenn sie rund um die Uhr im Einsatz sind und parallelisiert werden, um somit einen annähernd gleichen Durchsatz zu erreichen, wie ihn ein Mitarbeiter in kürzerer Zeit erzielen könnte.

Eine Analyse von Kundenanfragen aus dem Jahr 2022 (Spenrath 2022) ergab, dass bei einer Prozesstaktzeit von zehn Sekunden nur maximal 40% der Anwendungen realisiert werden können. Beträgt die Prozesstaktzeit bereits 20 Sekunden, sind nur noch 19% der Anfragen realisierbar. Diese Analyse zeigt die Relevanz von niedrigen Taktzeiten, welche im Allgemeinen durch industrielle Produktionsprozesse vorgegeben werden.

Da es sich bei Vereinzelungsanwendungen selten um vollständig repetitive Aufgaben handelt, der Prozess von Bildsensordaten abhängt und kein Prozesszyklus exakt dem vorherigen gleicht, treten Prozessfehler häufiger auf. In Spenrath (2022) wurde die Leistung



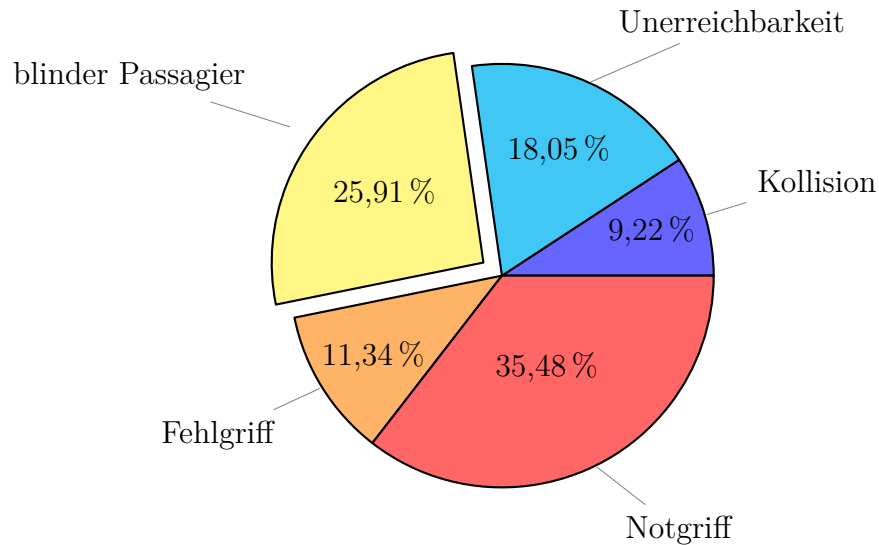


Abbildung 1.3: Fehleranalyse im Dauertest von zwei Roboteranlagen zur Vereinzelung von Pleuelstangen nach (Spennrath 2022)

zweier Griff-in-die-Kiste-Roboteranlagen zur Vereinzelung von Pleuelstangen untersucht. Basierend auf den dabei erfassten Daten lassen sich detaillierte Analysen der fehlerhaften Griffe durchführen. Insgesamt wurden 1.556.077 Griffe betrachtet. Wie in Abbildung 1.2 dargestellt, sind 70,62% der Griffe erfolgreich durchgeführt worden. 29,38% der Griffe sind fehlgeschlagen. Bei einer Produktivität von ungefähr 70% sind Puffer und Zwischenspeicher notwendig, um die Taktzeit zu entkoppeln und den Produktionsprozess nicht zu unterbrechen. Diese Pufferlösungen sind kostenintensiv und machen durch ihre Notwendigkeit eine Anlage häufig unwirtschaftlich.

In dieser Arbeit wird eine Methode präsentiert, die die Produktivität von Griff-in-die-Kiste-Applikationen für verhakende Werkstücke, wie beispielsweise Pleuelstangen, steigert und damit die Anlagenverfügbarkeit wesentlich erhöht.

## 1.2 Problemstellung

Wie bereits Abbildung 1.2 zeigt, sind in zwei beispielhaften Griff-in-die-Kiste-Roboteranlagen zur Vereinzelung von Pleuelstangen rund 30% der durchgeführten Griffe fehlerhaft. In Abbildung 1.3 sind die Ursachen und Häufigkeiten dieser fehlerhaften Griffe dargestellt.

Die geringste Ausfallrate von 9,22% wird in der Analyse durch Kollisionen hervorgerufen. Kollisionen können während des Prozesses unter anderem zwischen dem Roboterarm, dem angeflanschten Greifer, der Kiste und durch verrutschte Werkstücke in der Kiste auftreten. Mit 11,34% werden Fehlgriffe detektiert. Ein Fehlgriff kann detektiert werden, wenn beispielsweise die Öffnungsweite des Greifmoduls überwacht wird. Wenn die tatsächliche Öffnungsweite nicht mit der erwarteten Öffnungsweite übereinstimmt, schlägt der Griff fehl. Dieser Fehler kann durch falsche Roboterposen, ungenaue Hand-Auge-Kalibrierung, fehlerhafte Objektlageerkennung oder durch zu nahe aneinanderliegende Werkstücke auftreten, welche zwischen den Greiferbacken eingeklemmt werden können.

Abhängig von der Werkstück- und Greifergeometrie sind nicht alle Werkstücklagen in einer Kiste greifbar. Wenn kein stabiler Griff am Werkstück möglich ist, kann ein instabiler Griff, ein sogenannter Notgriff, durchgeführt werden. Nach einem solchen Notgriff muss die Werkstücklage erneut erfasst und somit ein neuer Prozesszyklus gestartet werden. Das häufigste Auftreten von Notgriffen wird bei sehr geringen Kistenfüllgraden beobachtet. Dennoch kommt es bei einem Notgriff zu keinem Anlagenstillstand, sondern lediglich zu einer doppelten Prozesszyklusdauer. In der Analyse wurden 35,48% der erfassten Prozessfehler durch Notgriffe ausgelöst. In 18,05% der Fehlerfälle haben trotz Notgriffe keine erfolgreichen Vereinzlungen stattgefunden. Sie wurden demnach mit Unerreichbarkeit annotiert.

Blinde Passagiere sind Werkstücke, die zusätzlich zu dem zum Greifen vorgesehenen Werkstück mittransportiert werden. Dies kann unterschiedliche Gründe haben. Beispielsweise liegen die gefundenen Werkstückgeometrien kongruent übereinander, die Werkstücke haben an der zu greifenden Stelle eine ähnliche Geometrie und ein Griff wurde zu tief berechnet. Dieser Fall tritt allerdings sehr selten ein. Ein viel größerer Anteil der blinden Passagiere wird durch verhakungsfähige Werkstückgeometrien in der Kiste verursacht. Bei einer Pleuelstange kann der Pleuelkopf im größeren Pleuelauge stecken und somit beim Anheben eines der Werkstücke eine Verhakung verursachen, wie in Abbildung 1.4 dargestellt. Mit 25,91% stellen Verhakungen und blinde Passagiere somit den zweitgrößten Fehleranteil der analysierten Griff-in-die-Kiste-Applikation und damit bei Reduzierung oder Eliminierung, die größte Effizienzsteigerung dar.

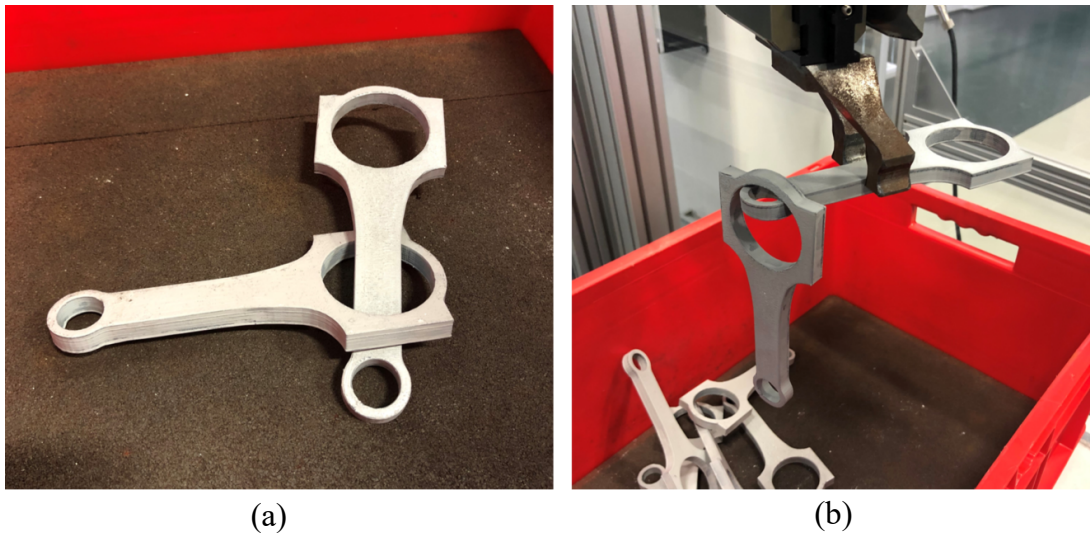


Abbildung 1.4: Verhakte Werkstücksituation von Pleuelstangen in der Kiste (a) und nach einem Griff als blinder Passagier (b)

Oft werden bei der Vereinzelung verhakender Werkstückgeometrien sicherheitshalber zusätzliche Enthakungsbewegungen durchgeführt, die im Falle einer Verhakung die Bauteile voneinander trennen. Ohne eine automatische Erkennung von Verhakungen müssen diese Bewegungen allerdings nach jedem Griff aus der Kiste ausgeführt werden und sind nicht auf die verhakte Situation abgestimmt, was die effektive Taktzeit des Vereinzelungsprozesses enorm verlängert. Zudem müssen diese zusätzlichen Bewegungen auf die jeweiligen Werkstückgeometrien aufwendig durch einen Experten angepasst werden, was entgegen einer flexiblen Lösung spricht. Das Abfallenlassen zurück in die Kiste, als weitere Strategie zur Enthakung, ist für sensible Werkstücke meist nicht zuträglich und nicht erlaubt.

Eine weitere Methode, die Fehlerquote von blinden Passagieren zu reduzieren, sind sensorische Überwachungen. Kraft-Momenten-Sensoren (KMS) können zwischen Greifer und Roboterarm angebracht werden, um auftretende Kräfte und Momente zu überwachen, welche durch einen zusätzlich angehobenen blinden Passagier ausgelöst werden. Je kleiner und leichter die zu vereinzelnde Werkstückgeometrie ist, desto schwieriger ist es allerdings, hiermit robuste Detektionen von blinden Passagieren zu gewährleisten. Das Gewicht des Greifers muss je nach Art der Montage des KMS mitberücksichtigt werden. Ferner sind zur Erkennung von blinden Passagieren optische Sensoren, wie beispielsweise Lichtschranken,

im Einsatz. Nachdem das Werkstück aus der Kiste bewegt wurde, wird auf dem Weg zur potenziellen Ablagestelle eine Lichtschranke durchfahren, welche nur dann auslöst, wenn etwas unterhalb des gegriffenen Werkstücks den Lichtstrahl durchbricht. Dies setzt allerdings einen robusten Griff voraus, ohne dass sich das Werkstück im Greifer bewegt. Zudem sollte der blinde Passagier tiefer als das gesamte gegriffene Werkstück im Greifer hängen. Eine robuste Methode zur Detektion von blinden Passagieren ist eine zusätzliche Bildverarbeitung. Der Roboterarm hält mit dem Greifer das gegriffene Werkstück vor eine zwei- oder dreidimensionale Bildsensorik zur Generierung eines Abbildes, welches in einem nachgelagerten Schritt ausgewertet wird. Auch hierbei kommt es zu Verlängerungen der Taktzeit des Vereinzelungsprozesses. Zudem ist mit weiteren Kosten für die Anschaffung und Instandhaltung eines Bildverarbeitungsprozesses zu rechnen. Als Fazit kann geschlossen werden, dass alle genannten Ansätze den Nachteil aufweisen, dass eine Verhakung erst nachträglich nach dem Griff erkannt wird und sich die Taktzeit durch das Zurücklegen oder den Versuch einer Abwurfbewegung deutlich verlängert.

Eine Lösung zur intelligenten, effizienten und wirtschaftlichen Verhakungserkennung und zur Enthakung von verhakten Werkstückgeometrien in Griff-in-die-Kiste-Applikationen wird in dieser Arbeit vorgestellt.

### **1.3 Stand der Forschung**

Die Forschung rund um das Thema des Handhabens und Vereinzelns von Produkten und Werkstücken aller Art ist durch den Einsatz maschineller Lernmethoden in den letzten Jahren enorm gewachsen. Ein wichtiger Bestandteil dieser Forschung ist die Skalierbarkeit und Robustheitssteigerung von diesen Systemen. Neben modellfreien und modellbasierten Ansätzen und Möglichkeiten der Beschleunigung einzelner Teilprozesse, sind auch Ansätze zur automatischen Verhakungserkennung und Enthakung entstanden. In diesem Kapitel werden einige dieser Ansätze vorgestellt und kritisch beleuchtet. Hierbei wird zwischen modellfreien und modellbasierten Ansätzen unterschieden. Zu Beginn eines jeden Unterabschnitts wird kurz auf die Entwicklung von Methoden zur allgemeinen Werkstückvereinzelung eingegangen, bevor der Fokus auf die Verhakungserkennung und Enthakung gelegt wird.

### 1.3.1 Modellfreie Ansätze

Modellfreies Greifen erlangt in der aktuellen Zeit immer mehr Beachtung. Durch kleine Losgrößen besteht immer mehr Bedarf an Vereinzelungslösungen, die kaum bis keinen Einrichtungsaufwand durch Experten benötigen. Beim modellfreien Greifen liegt dem System keinerlei Information über die Geometrie des zu vereinzelnden Werkstücks vor. In Tiefenbildern der Kistensituation werden keine Werkstücklagen lokalisiert, sondern mögliche Greifposen und -flächen gesucht. Da mit diesen Ansätzen möglichst viele unterschiedliche Werkstücke gehandhabt werden sollen, erhält diese Technologie immer mehr Einzug in die Intralogistik, den Versandhandel und den Einzelhandel.

#### Greifplanung

Ein erster geometrisch analytischer Ansatz durch den Einsatz von Optimalfiltern von Dessimoz et al. (1984) erfolgt zweidimensional. In RGB-Bildern werden Strukturen gesucht, welche sowohl von Klemm- als auch von Sauggreifern vereinzelt werden können. Ein weiterer geometrischer Ansatz bildet aus zwei dreidimensionalen Abbildern konvexe Geometrien, an welchen für Klemmgreifer parallel gegenüberliegende Greifflächen gesucht werden (Trobina 1995). Bone et al. (2008) nutzt vier Kamerapositionen, um aus den zusammengesetzten Daten Oberflächen zu finden. Anhand dieser Oberflächen werden Normalenvektoren aufgespannt und mit Berücksichtigung der Distanz dieser Oberflächen mögliche Greifposen gesucht. Berger et al. (2000) sucht in der Tiefeninformation nach der höchstgelegenen, größtmöglichen Fläche, während Richtsfeld & Vincze (2008) unter Berücksichtigung des Massenschwerpunkts nach Greifflächen sucht. Kimitoshi et al. (2010) teilt das lokalisierte Werkstück imaginär durch eine Ebene und versucht in dieser mögliche Greifposen zu finden. Erste Ansätze, welche die Geometrie des Greifers mitberücksichtigen, werden von He et al. (2021) und Domae et al. (2014) vorgestellt. In beiden Ansätzen wird versucht, die Geometrie der Greiferfinger auch in Form von Masken in die dreidimensionalen Tiefenbilder einzupassen. Um unterschiedliche Werkstückgeometrien robust vereinzeln zu können, ist es ebenfalls notwendig, unterschiedliche Greifertypen und -methoden einsetzen zu können. Vermehrt wird mit Systemen mit Klemmgreifern und Sauggreifern gearbeitet (Khalid et al. 2021). Auch universellere Greifmethoden halten hier Einzug, wie

beispielsweise ein *FlexShapeGripper* der Firma Festo Vertrieb GmbH & Co. KG<sup>1</sup> oder eine *Formhand* der Firma Formhand Automation GmbH<sup>2</sup> (Löchte et al. 2014). Letztere arbeitet mit Volumenstromevakuierung aus mit Granulat gefüllten Kissen, welche sich an sämtliche Oberflächen der Werkstücke im Prozess anpassen können (Kunz et al. 2015). You et al. (2018) stellt zudem einen Ansatz mit anhand von Tiefenbildern relativ zueinander verschiebbaren Sauggreifern vor. Hierbei wird zuerst die Tiefeninformation segmentiert, die Oberfläche projiziert und dann ein Mustervergleich verschiedener Greiferkonstellationen ausprobiert, um eine geeignete Greifpose mit einem translatorisch variablen Sauggreifer zu finden. Zeng et al. (2022) arbeitet mit einem Saug- und Klemmgreifer, welcher durch die Prognose von zwei neuronalen Netzwerken ausgewählt wird.

Der Einsatz maschineller Lernverfahren zur Prognose möglicher Greifposen im Tiefenbild ist von Pauli (1998) etabliert. Durch manuelles Bewegen des Greifers zur potenziellen Greifpose und durch die dadurch gestartete Aufnahme von Referenzbildern der Werkstück- und Greifergeometrie, wird ein neuronales Netzwerk trainiert, welches auf ungesehene Situationen generalisiert. Lin & Sun (2015) trainiert neuronale Netze anhand Trainingsdaten, die durch menschliches Vormachen der zu bewältigenden Aufgabe aufgezeichnet werden. Hierbei wird nicht nur die Greifpose an sich, sondern auch die Anfahrt- und Abfahrttrajektorie des Griffes berücksichtigt. Gleiches manuelles Einlernen wird von Aleotti & Caselli (2008), Argall et al. (2009) und Ekvall & Kragic (2004) erforscht. El-Khoury & Sahbani (2010) fokussiert zudem auf die Aufgabenabhängigkeit, d.h. für welchen Zweck ein möglicher Griff ausgeführt werden muss. Dies erlaubt weitere Prognosen und Vorbereitungen für nachfolgende Prozesse. Lenz et al. (2015) bringen zwei neuronale Netzwerke für annotierte RGB-D-Daten zum Einsatz, welche kaskadiert zuerst grobe Prognosen möglicher Greifposen im Tiefenbild liefern, bevor das zweite Netzwerk detaillierter danach sucht. Mahler et al. (2017) suchen auf der Basis von Punktwolken nach Greifposen. Mit einem Datensatz von 6,7 Millionen synthetisch generierten Tiefenkarten von CAD-Modellen mit der jeweiligen annotierten Greifpose wird ein Training eines neuronalen Netzwerkes durchgeführt, welches mögliche Greifposen schätzt. Die Greifposen werden automatisch mit

---

<sup>1</sup><https://www.festo.com/>

<sup>2</sup><https://www.formhand.de/>

Klemm- und Sauggreifern an virtuellen Werkstückmodellen erzeugt (vgl. Johns et al. 2016). Dune et al. (2008) nutzen vereinfachte kubische Volumenkörper zur Nachbildung möglicher Werkstücke, um an diesen mögliche Greifposen zu finden. Huebner & Kragic (2008) setzt zudem eine Heuristik ein, um Greifposen zu finden. Aufgrund von Orientierungsproblemen und Formproblemen beim Finden der Posen wird ein neuronales Netz nachgeschaltet, um eine entgültige Greifpose auszuwählen. Morrison et al. (2018) nutzen einen vierdimensionalen Griff, welcher durch ein überwachtes Lernverfahren anhand von 885 Objekten im annotierten Datensatz prognostiziert wird. Nebenbei schätzt das überwachte Lernverfahren die Qualität des vorgeschlagenen Griffes und die Öffnungsweite des Klemmgreifers.

Entgegen des überwachten Lernverfahrens, welches mit annotierten Trainingsdaten arbeitet, existieren auch modellfreie Vereinzelmethode anhand von bestärkenden Lernverfahren (vgl. Shao et al. 2019). Das System wird bei erfolgreichen Teilprozessen belohnt und bei fehlerhaften bestraft. Levine et al. (2018) lernen die Hand-Auge-Koordination anhand von RGB-Bildern, die durch Kameras über 14 Ladungsträger aufgezeichnet werden. Aus diesen Ladungsträgern vereinzeln parallel 14 Roboter in rund 800.000 Greifversuchen Objekte. Hierdurch generalisiert das System auf ungesehene Objekte und korrigiert Fehler anhand von *Continuous Servoing*. Kalashnikov et al. (2018) nutzen ebenfalls bestärkendes Lernen und fokussieren auf die Vormanipulation der Objekte im Ladungsträger, beispielsweise durch das Verschieben in der Kistenbodenebene (vgl. Berscheid et al. 2019). Durch den Einsatz eines Perzeptionsnetzwerkes von Zeng et al. (2020) werden Objekte zuerst vorklassifiziert, bevor die RGB-D Daten in ein Greif- und Wurfnetzwerk eingespeist werden. Der *Tossing Bot* ist in der Lage, physikalische Eigenschaften des Objektes durch alleinige Erkennung im Ladungsträger zu prognostizieren und die gegriffenen Objekte in vordefinierte Regionen abzuwerfen.

### **Verhakungserkennung**

Eine modellfreie Methode zur Verhakungserkennung von chaotisch gelagerten Werkstücken in Ladungsträgern wird von Matsumura et al. (2019) vorgestellt. Das Tiefenbild wird auf die prognostizierte Greifpose zugeschnitten und einem neuronalen Netzwerk übergeben. Dieses prognostiziert die Wahrscheinlichkeit einer Verhakung und übergibt diese Infor-

mation der Greifplanung. Gleiches erzielt Zhang et al. (2021) durch die Erzeugung einer topologischen Karte, welche beinhaltet, in welchem Bereich des gesamten Tiefenbildes mögliche Verhakungen auftreten können. Hierbei wird sich dem *Gauß'schen Linking Integral* beholfen, welches die mögliche Windung und damit Verhakung der Werkstücke in der vorliegenden Szene analysiert. Zudem wird die regionale Anhäufung von Verhakungssituationen in der Kiste prognostiziert. Aufgrund der Modellfreiheit sind beide Methoden nur für eine Verhakungserkennung, nicht aber für eine intelligente Enthakungslösung geeignet. Nur durch Kenntnis der Geometrie der verhakten Werkstücke kann intelligent enthakt werden, da anhand dieser Geometrie Enthakungstrajektorien geplant werden können.

### 1.3.2 Modellbasierte Ansätze

Modellbasierte Ansätze zur Objektlageerkennung und Greifplanung werden zumeist in Produktionsprozessen eingesetzt, da hiermit eine lagerichtige Ablage der gegriffenen Werkstücke realisierbar ist. Bei modellbasierten Vereinzelungslösungen von Schüttgut aus einem Ladungsträger spricht man allgemein von Griff-in-die-Kiste-Applikationen.

#### Griff-in-die-Kiste-Methoden

Zu Beginn jedes Vereinzelungsprozesses muss die Kistensituation durch einen meist dreidimensionalen Bildsensor erfasst und mögliche bekannte Werkstückgeometrien lokalisiert werden. Hierfür sind viele Ansätze bekannt (Ledermann 2012; Palzkill & Verl 2012; Effenberger et al. 2013; Kleeberger & Huber 2020; Dong et al. 2019). Konishi et al. (2019) setzt hierbei beispielsweise auf RGB-D-Daten und nutzt Berechnungen auf der *central processing unit* (CPU). Die Methoden werden in objektbasierte, ansichtenbasierte Verfahren und maschinelle Lernverfahren unterteilt. Objektbasierte Methoden suchen nach vollständigen Objektbeschreibungen in der Tiefeninformation, wie beispielsweise Kanten, Kurven, Oberflächen oder ganze Volumen. Ansichtenbasierte Verfahren arbeiten mit Datenbanken der Ansichten, welche vor dem Lokalisierungsprozesses generiert werden. Primär suchen diese Verfahren nach Formen, Merkmalen und Strukturbeschreibungen der bekannten Werkstücke. Maschinelle Lernverfahren nutzen beispielsweise eine Grid-Methode, welche ein Gitternetz über ein Tiefenbild spannt und die Wahrscheinlichkeit von Werkstückursprüngen in einem Raster schätzt.



Nach einer erfolgreichen Objektlageerkennung werden mögliche Greifposen am Werkstück analysiert, welche vorab manuell definiert (Luo et al. 2015; Papazov et al. 2012; Chen et al. 2018) oder automatisch generiert werden können (Dupuis et al. 2008; Johns et al. 2016; Goldfeder et al. 2007; Sanz et al. 2005; Miller et al. 2003; Khalid et al. 2021; Kleeberger et al. 2021). Bohg & Kragic (2010) versucht anhand des Geometriekontextes mögliche Greifposen am Werkstück zu erlernen. Um die generierten und gefundenen Greifposen an den Werkstücken in der Kiste zu erreichen, fährt der Roboter mit Greifer Anfahr- und Abfahrbahnen ab. LaValle & Kuffner Jr (2001) nutzt *Rapidly Exploring Random Trees*, kurz RRTs, um dynamische Pfadplanungen zu vollziehen. Hierbei wird ein Baum möglicher Pfade aufgespannt, welcher inkrementell in Richtung der Zielkonfiguration wächst und dabei durch die weniger erforschten Bereiche beeinflusst wird. Berenson & Srinivasa (2008) sucht anhand bidirektionaler RRTs nach möglichen Anfahrts- und Abfahrtsbahnen und nutzt zur Greifposendetektion kraftgeregelt Greifmodule. Ferrari & Canny (1992) hingegen nutzt einen Zweifingergreifer und untersucht den Kraftschluss, um eine optimale Greifpose zu finden. Für Roboterbahnen zur Entnahme der Werkstücke ist es notwendig, Kollisionsprüfungen mit den Werkstücken, sowie zwischen Roboterarm, Greifergeometrie und Ladungsträger durchzuführen. Harada et al. (2013) prüfen den Greifer entlang der Roboterbahn mithilfe eines Zylinders als vereinfachte Greifergeometrie. Weitere Ansätze überprüfen mögliche Kollisionen mit zweidimensionalen Masken der Greiferfinger eines Klemmgreifens (Martinez et al. 2015) oder mit *Bounding Boxen* der Werkstücke (Takenouchi et al. 1998). Vollständige Kollisionsüberwachungen mit der Punktwolke des Tiefenbildes werden von Schyja et al. (2012) durchgeführt. Es bestehen zudem Ansätze zur iterativen Bahnüberwachung von und bis zur jeweiligen Greifpose am Werkstück (Spenrath et al. 2012; Lin et al. 1997). Eine vollständige Kollisionsprävenz von Greifern, Werkstücken, Robotern, Ladungsträgern und weiteren Gegenständen in der Roboterzelle sind für einen robusten Griff-in-die-Kiste-Prozess notwendig.

### **Methoden zur Verhakungserkennung und Enthakung**

Einen modellbasierten Ansatz zur Verhakungserkennung und zur Enthakung von biegeschlaffen Rohren stellen Leão et al. (2020) und Leão et al. (2022) vor. Hierzu werden

zuerst die Rohre mit kleinen geraden Rohrstücken als CAD-Modell nachgebildet, welche dann mit einer Objektlageerkennung in der Werkstückanhäufung lokalisiert werden. Nach erfolgreicher Lokalisierung werden die Rohrstücke durch eine Kostenfunktion auf ihre Greifbarkeit und Vereinzelbarkeit bewertet. Die Kostenfunktion setzt sich aus der Höhe des gefundenen Werkstücks relativ zur höchsten Werkstücklage in der Kiste, der Distanz des gefundenen Teilwerkstücks zum nächstgelegenen Ende des Rohres und der prognostizierten minimalen Verschiebung anderer Werkstücke im Ladungsträger, sobald das Rohr angehoben wird, zusammen. Diese Metrik erlaubt eine Präferenz zum Greifen des am wenigsten verhakten Rohrs. Zum Enthaken verwenden Leão et al. (2020) und Leão et al. (2022) eine Distanzberechnung des gefundenen Rohrstücks zum nächstgelegenen Ende des Rohres und verfährt den Greifer parallel zur Kistenbodenebene, um diese Distanz zu überwinden und das gegriffene Rohr freizulegen. Durch einen Kraft-Momenten-Sensor (KMS) wird der Prozess überwacht und verhakte Werkstücke nach erfolgloser Enthakung wieder zurück in den Ladungsträger geworfen. Ein weiterer Ansatz wird von Ray & Howard (2020) vorgestellt. Zur Entwirrung von Salatblättern werden große Salatblattanhäufungen erkannt und durch Spreizen eines Klemmgreifers getrennt, bevor mittels einer erneuten Lokalisierung gegriffen werden kann.

Die Erforschung von Methoden zur automatisierten Verhakungserkennung und Enthakung ist ein noch sehr neues und unerforschtes Thema, welches neben dem Autor nur von sehr wenigen Wissenschaftlern erforscht wird.

### **1.3.3 Schlussfolgerung**

In Produktionsprozessen werden Werkstücke meist orientiert und lagerichtig vereinzelt benötigt. Dies ist nur mit modellbasierten Methoden möglich, da hier die Werkstückgeometrie bekannt ist und die Ablage nach einem erfolgreichen Griff geplant werden kann. Modellfreie Ansätze finden hingegen beispielsweise in Warenlagern und bei Kommissionieraufgaben Anwendung (Bormann et al. 2019). Modellfreie Ansätze zur Verhakungserkennung generalisieren auf ungesehene Werkstückgeometrien und benötigen deshalb kaum bis keinen Einrichtungs- und Rüstaufwand (Matsumura et al. 2019; Zhang et al. 2021). Um allerdings intelligente Enthakungsstrategien zu entwickeln, muss die Werkstückgeometrie bekannt

sein, um Enthakungsbewegungen relativ zu dieser Geometrie planen und ausführen zu können. Existierende Ansätze zur Verhakungserkennung, die modellbasiert sind, sind meist nur auf eine bestimmte Werkstückgeometrie ausgelegt und funktionieren nach analytischen Methoden, was eine Generalisierung auf ungesehene Werkstückgeometrien erschwert (Leão et al. 2020; Leão et al. 2022). Enthakungsmethoden, die bislang eingesetzt werden, sind ebenfalls stark auf eine bestimmte Geometrie des Werkstückes ausgelegt und mögliche Enthakungsbewegungen werden nur in einer Ebene parallel zur Kistenebene ausgeführt. Für viele Verhakungsgeometrien reicht eine solche Bewegung nicht aus, um die Werkstücke voneinander trennen zu können. Auch ein aktives Verschieben der Werkstücke, um Verhakungen zu vermeiden, ist in industriellen Produktionsprozessen aufgrund von möglichen Beschädigungen am Bauteil meist nicht gewünscht. Ein erster allgemeiner modellbasierter Ansatz, Werkstückverhakungen zu detektieren und intelligent zu trennen, welcher für alle möglichen Werkstückgeometrien eingesetzt werden kann, wird in dieser Arbeit entwickelt.

## 1.4 Zielsetzung und Abgrenzung

Diese Arbeit hat das Ziel, eine Methode zur Verhakungserkennung und Enthakung von verhakten Werkstücken in einer Griff-in-die-Kiste-Applikation zu entwickeln und zu untersuchen. Die Verhakungserkennung soll bereits in der Kiste, durch die Verwendung einer vorhandenen Punktwolke und der durch eine Objektlageerkennung zur Verfügung gestellten Werkstückposen, alle gefundenen Werkstücke auf eine mögliche Verhakung überprüfen. Die Objektlageerkennung (Palzkill 2014) wird als gegeben angenommen.

Die Enthakungsmethode soll nach erfolgreicher Detektion einer Verhakung den durch eine Greifplanung vorgegebenen Entnahmepfad durch zusätzliche Pfadposen zur Enthakung erweitern und somit die Werkstücke bereits während der Entnahme in der Kiste voneinander trennen. Hierfür wird eine bestehende heuristische Greifplanung (Spenrath 2022) verwendet, welche erweitert wird. Die Kollisionsvermeidung ist Bestandteil der verwendeten Greifplanung. Die Kenntnis über die zu einzelnden Werkstücke und die geometrischen Daten in Form von CAD-Modellen werden vorausgesetzt. Zudem wird diese Arbeit für

starre Werkstückgeometrien in sortenreinen Ladungsträgern abgegrenzt. Vorab definierte Greifposen am Werkstück werden in dieser Arbeit als gegeben angenommen.

Die Griff-in-die-Kiste-Applikation muss verschiedene Anforderungen erfüllen, um Markt- und Prozessbedürfnisse zu befriedigen. Die Methoden zur Erkennung und Trennung von Verhakungen sollen in der Lage sein, diesen Anforderungen gerecht zu werden. Insbesondere werden Effizienz, Skalierbarkeit und Flexibilität von solchen Methoden zur Robustheitssteigerung erwartet, um den Prozess zu verbessern.

### **Anforderung: Effizienz**

Die Effizienz einer Roboteranlage vereint ihre Robustheit und Verfügbarkeit. Die Verhakungserkennung und Enthakungsmethode sollen die Effizienz einer Griff-in-die-Kiste-Roboteranlage steigern und die effektive Taktzeit durch Reduzierung von Stillstandzeiten verkürzen. Beispielsweise sollen nicht verhakte Werkstücksituationen in der Kiste zum Greifen präferiert und nur dann eine Enthakungstrajektorie abgefahren werden, wenn eine verhakte Situation vorliegt. Die Anzahl zusätzlicher Pfadposen zu einer Trennung der verhakten Werkstücke soll intelligent auf die verhakte Werkstücksituation angepasst sein. Die minimale Anzahl zusätzlicher Pfadposen zur Trennung soll eingesetzt werden.

### **Anforderung: Skalierbarkeit**

Die Methoden zur Verhakungserkennung und Enthaken sollen schnell auf neue Werkstückgeometrien adaptierbar und damit skalierbar sein. Die Parametrierungen der Methoden sollen fast vollständig automatisiert in einer Simulationsumgebung durchgeführt werden, sodass wenig manueller Parametrieraufwand besteht und dieser von keinem Experten durchgeführt werden muss. Als zusätzliches Modul für Griff-in-die-Kiste-Softwarelösungen soll es nach Bereitstellung der benötigten Daten verwendbar sein.

### **Anforderung: Flexibilität**

In produzierenden Unternehmen werden sehr oft unterschiedliche Ladungsträger und Kistensysteme verwendet, weshalb das Verhakungserkennungs- und Enthakungsmodul mit einer beliebigen Ladungsträgerart und -größe einsetzbar sein soll. Zudem werden für

unterschiedliche Werkstückgeometrien oft spezielle Greifer entwickelt und konstruiert, um diese robust handhaben zu können oder die Erreichbarkeit in der Kiste zu erhöhen. Das entstehende Modul soll auch hierfür eine allgemeine Lösung bieten, welche sowohl mit Klemm- und Sauggreifern, als auch mit weiteren Greiferarten Verhakungen trennen kann. Verschiedene Eingangsdaten sind für ein solches Modul notwendig, welche durch unterschiedliche Objektlageerkennungsmethoden zur Verfügung gestellt werden können.

Schlussfolgernd soll in dieser Arbeit folgende Forschungsfrage beantwortet werden:

*Wie kann eine Situation von verhakten Werkstücken mit formfester, bekannter Geometrie und mit vordefinierten Greifposen in einem sortenreinen Ladungsträger beim Griff-in-die-Kiste effizient erkannt und die verhakten Werkstücke nach dem Greifen durch einen Industrieroboter effizient getrennt und vereinzelt werden?*

## 1.5 Lösungsansatz und Gliederung der Arbeit

In dieser Arbeit wird ein Verfahren entwickelt, welches verhakte Werkstücksituationen in sortenreinen, mit Schüttgut gefüllten Kisten, erkennt. Zudem werden für ein mögliches Verfahren zur Enthakung zwei überwachte Ansätze und ein bestärkender Ansatz des maschinellen Lernens vorgestellt und verglichen, welche alle einem einheitlichen Hemisphärenmodell von Bewegungsbahnen zu einer Enthakung folgen.

In Kapitel 2 werden zunächst mathematische Grundlagen und Notationen eingeführt, die zum Verständnis dieser Arbeit notwendig sind. Im Anschluss wird die Methode zur automatisierten Werkstückvereinzlung durch Industrieroboter erläutert. Nach einer Vorstellung notwendiger Hardwarekomponenten einer Griff-in-die-Kiste-Applikation wird in die Algorithmik der Griff-in-die-Kiste-Software, sowie in die Objektlageerkennung und die Greifposengenerierung eingeführt. Ferner werden zudem die Greifposensuche und die Entnahmepfadplanung mit Kollisionserkennung beschrieben, da dieses Softwaremodul eine wesentliche Rolle in der Entwicklung einer Verhakungserkennungs- und Enthakungslösung spielen. In maschinelle Lernverfahren wird durch Darstellungen von tiefen neuronalen

Netzen, überwachen und bestärkenden Lernverfahren eingeführt. Zudem wird der Einsatz maschineller Lernverfahren in der Handhabungsautomatisierung dargestellt. Es werden ebenfalls Transfermethoden beschrieben, welche maschinelle Lernverfahren von Simulationsumgebungen in die reale Welt übersetzen. Die Beschreibung einer Umgebung zur Simulation von Vereinzlungsprozessen zur Generierung von Daten und zum Training maschineller Lernverfahren runden Kapitel 2 ab.

In Kapitel 3 wird ein Verfahren zur Verhakungserkennung von Werkstücksituationen in einem Ladungsträger beim Griff-in-die-Kiste nach Moosmann et al. (2020) entwickelt. Nachdem die Ausgangslage vorhandener Daten und deren Aufbereitung beschrieben wird, folgt eine Vorstellung einer neuronalen Netzarchitektur. Folgend wird das Training dieser neuronalen Netzarchitektur und ein anschließender Transfer von der Simulationsumgebung in die reale Welt nach Moosmann et al. (2022b) erläutert. Am Ende des Kapitels wird die entwickelte Methode in das Griff-in-die-Kiste-Gesamtsystem eingeordnet.

In Kapitel 4 wird ein Verfahren zur Enthakung von verhakten Werkstücken nach Moosmann et al. (2021b) entwickelt. Zu Beginn wird die Ausgangslage vorhandener Daten und deren Aufbereitung beschrieben. Es wird ein Hemisphärenmodell zur Repräsentation von Bewegungsbahnen zur Enthakung vorgestellt, welches als Vergleichsgrundlage für verschiedene Lernverfahren dienen kann. Im Anschluss werden verschiedene Parameter des Modells in einer Versuchsreihe validiert und optimiert. Zum Abschluss der Entwicklung des Hemisphärenmodells werden zusätzliche Orientierungsänderungen an den Hemisphärenposen eingeführt. Um geeignete Pfadposen des Hemisphärenmodells zu prognostizieren, werden zwei überwachte und ein bestärkendes Lernverfahren (Moosmann et al. 2021a) vorgestellt. Hierbei wird zunächst ein Überblick über die Methoden gegeben sowie auf die Trainingsabläufe der jeweiligen neuronalen Netzwerkarchitekturen sowie eines Agenten eingegangen. Wie bereits bei dem Verfahren zur Verhakungserkennung wird eine Transfermethode von der Simulationsumgebung in die reale Welt erläutert (Moosmann et al. 2022b). Zuletzt wird ein Ablehnungskriterium zur Reduzierung unmöglicher Enthakungsperipherie, wie beispielsweise Greifposen am Werkstück, die eine Trennung der Verhakung verhindern,

---

vorge stellt (Moosmann et al. 2022a) und das Enthakungsverfahren in das Gesamtsystem eingeordnet und zusammengefasst.

In Kapitel 5 werden experimentelle Ergebnisse für die in dieser Arbeit entwickelten Verfahren vorge stellt. Nach der Beschreibung der Evaluierungsumgebungen in der Simulation und der Realität werden sowohl die Verhakungserkennung als auch die unterschiedlichen Methoden der Enthakung nach einem Simulationstraining gegenüber gestellt und nach einem direkten Transfer in einer realen Roboterzelle evaluiert. Zudem erfolgt ein Vergleich unterschiedlicher Methoden des Transfers von der Simulation in die Realität für die Verhakungserkennungs- und Enthakungsmethode. Das Kapitel endet mit einer Validierung des Gesamtsystems und des Effekts der in dieser Arbeit entwickelten Methoden.

In Kapitel 6 wird die Arbeit zusammengefasst und ein Ausblick auf mögliche Weiterentwicklungen des Gesamtsystems aufgezeigt.





## 2 Grundlagen

In diesem Kapitel werden Grundlagen vorgestellt, welche im Laufe dieser Arbeit Verwendung finden. Zunächst werden mathematische Grundlagen und Notationen eingeführt. Es folgt ein Abschnitt zum Einsatz maschineller Lernverfahren, welcher eine Einführung in die Thematik und Erläuterungen zum überwachten und bestärkenden Lernen beinhaltet. Zudem wird eine Methode zur statistischen Analyse einer binären Klassifizierung vorgestellt. In Abschnitt 2.3 folgt eine Erläuterung der Komponenten einer Griff-in-die-Kiste-Vereinzelungslösung. Es werden 3D-Bildsensoren und eine Objektlageerkennungsmethode vorgestellt. Ferner wird auf Industrieroboter, Greifsysteme, eine Methode der Greifplanung und auf zusätzliche Roboterzellenperipherie eingegangen.

### 2.1 Mathematische Grundlagen und Notation

Die Lage von Objekten im euklidischen Raum wird auch als Pose bezeichnet, welche die Position und Orientierung eines Objekts beschreibt. Die Beschreibung erfolgt bezüglich eines Koordinatensystems. Koordinatensysteme werden in dieser Arbeit mit einem  $K$  deklariert. Somit wird beispielsweise das Werkstückkoordinatensystem als  $K_W$  bezeichnet. In der Robotik wird zumeist ein Welt-Koordinatensystem  $K_0$  eingeführt, auf welches alle weiteren Koordinatensysteme abhängig definiert werden. Die Pose kann als homogene Transformationsmatrix

$$T = \begin{pmatrix} R & \vec{p} \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} r^{x_1} & r^{y_1} & r^{z_1} & p^x \\ r^{x_2} & r^{y_2} & r^{z_2} & p^y \\ r^{x_3} & r^{y_3} & r^{z_3} & p^z \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (2.1)$$

ausgedrückt werden. Diese besteht aus einer Rotationsmatrix  $R$  und einem Ortsvektor

$\vec{p}$ , der den Translationsanteil beschreibt. Eine Transformation  $T$  kann nach Craig (2006) mehrere Interpretationen annehmen.

Eine mögliche Interpretation einer Transformation besteht in der Beschreibung eines Koordinatensystems. Die Transformation  ${}^0T_W$  beschreibt beispielsweise das Koordinatensystem des Werkstücks  $K_W$  relativ zum Welt-Koordinatensystem  $K_0$ .

Eine weitere Interpretation besteht in der Umrechnung einer Pose in ein anderes Koordinatensystem. So wird beispielsweise eine Pose  ${}^Wp_1$  im Koordinatensystem eines Werkstücks  $K_W$  relativ zum Welt-Koordinatensystem  $K_0$  mit der Transformation  ${}^0T_W$  zu

$${}^0p_1 = {}^0T_W \cdot {}^Wp_1 \quad (2.2)$$

umgerechnet.

Zur Verarbeitung von Koordinatensystemen in Griff-in-die-Kiste-Softwareanwendungen werden diese meist durch homogene Transformationsmatrizen repräsentiert. Posen können auch als sechsdimensionale Vektoren dargestellt werden, wobei hierfür Konventionen zur Darstellung von Rotationen beachtet werden müssen (Siciliano et al. 2008). Ferner können zur Darstellung von Rotationen auch Quaternionen und Achswinkeldarstellungen verwendet werden.

## 2.2 Maschinelles Lernen

### 2.2.1 Einführung

Das maschinelle Lernen bildet als Teil des Forschungsfeldes der künstlichen Intelligenz ein bereits seit Jahrzehnten bestehendes Forschungsgebiet (Welsch et al. 2018). Durch die Entwicklung von immer leistungsstärkeren Computern, auf welchen die Algorithmen berechnet werden, wird ein starker Anstieg der Relevanz dieser Technologien verzeichnet (Richter 2019). Insbesondere die Weiterentwicklung von Grafikprozessoren (engl. Graphical Processing Unit - GPU) und deren Verwendung für das Training von künstlichen, neuronalen

Netzwerken sind dafür verantwortlich. Die parallelen Architekturen der Grafikprozessoren und die Eigenschaft, große Datendurchsätze zu verarbeiten, ermöglichen eine signifikante Beschleunigung für das Training dieser Netzwerke.

Der Begriff der künstlichen Intelligenz wurde erstmals von McCarthy et al. (1955) definiert und beschreibt Maschinen, die in ihrem Verhalten einer menschlichen Intelligenz ähnlich sind. Eine allgemeine Definition von Maschinellern Lernen wurde von Mitchell (1997) geprägt. Diese besagt, dass wenn ein System seine Leistung für eine bestimmte Aufgabe aufgrund gesammelter Erfahrungen verbessern kann, eine Art des maschinellen Lernens vorhanden sein muss. Gemeint ist dabei das Erkennen und Vorhersagen von Zusammenhängen, die aus hochdimensionalen Eingangsdaten stammen (Murphy 2012). Die Zusammenhänge können also durch statistische Modelle erfahren werden, ohne diese explizit zu programmieren. Generiert werden können diese Vorhersagen beispielsweise durch künstliche, neuronale Netzwerke. Diese sind in Anlehnung an das menschliche Gehirn konzipiert (Rey & Wender 2019). Die Struktur des bekanntesten künstlichen, neuronalen Netzwerktyps, dem Mehrschichtperzeptron (engl. Multilayer Perceptron - MLP) besteht aus einer Eingangsschicht, mehreren versteckten Schichten und einer Ausgangsschicht, beispielsweise bei einem vollständig verbundenen Netzwerk (engl. fully connected neural network). Die Eingangsschicht erhält dabei Daten aus der Umgebung oder aus einem Datensatz, verarbeitet diese und gibt sie an die versteckten Schichten weiter. Wenn zwei oder mehr versteckte Schichten im Netzwerk enthalten sind, spricht man von tiefen, neuronalen Netzwerken. Die letzte Schicht ist die Ausgangsschicht, welche die Prognosen des Netzwerks an die Umgebung übergibt.

Eine Schicht eines künstlichen, neuronalen Netzwerks besteht aus mehreren Neuronen. Ein künstliches Neuron ist einem biologischen Neuron nachempfunden und besitzt beispielsweise die Fähigkeit, auf Reize zu reagieren und Signale an weitere Neuronen weiterzuleiten. Hierbei erhält das künstliche Neuron die Ausgangswerte  $a_i$  der vorherigen Schicht, welche durch die Verbindungen mit einem Gewichtungsfaktor  $g_i$  multipliziert werden.

Zusätzlich zur Summe der gewichteten Ausgangswerte der vorherigen Schicht kann ein Bias  $b$  addiert werden. Der Bias kann den Schwellwert eines Neurons anheben. Die entstehende Summe

$$z = \sum_{i=0}^n (a_i g_i) + b \quad (2.3)$$

wird im nächsten Schritt durch eine Aktivierungsfunktion transformiert (Rey & Wender 2019). Die Aktivierungsfunktion verhindert eine lineare Abbildung des Ausgangssignals des Neurons und damit den Einsatz eines linearen Regressionsmodells. Die am häufigsten eingesetzten Aktivierungsfunktionen sind die ReLU-Funktion (engl. Rectified Linear Unit) oder die Sigmoid-Funktion (Goodfellow et al. 2016; Dong et al. 2020; Sutton & Barto 2018). Die Sigmoid-Funktion

$$f(z) = \frac{1}{1 + e^{-z}} \quad (2.4)$$

ist eine verschobene und skalierte Tangens-Hyperbolicus-Funktion (Dong et al. 2020), während die ReLU-Funktion für eine negative Eingabe den Wert null und für eine positive Eingabe den Eingabewert ausgibt.

Um das Training eines künstlichen, neuronalen Netzwerks zu steuern, wird die Abweichung der Prognose zum Soll-Wert bestimmt. Durch Fehlerrückführung (engl. Backpropagation) werden die Gewichte  $g_i$  des Netzwerks so angepasst, dass im Falle der gleichen Eingangsdaten eine Reduzierung der Abweichung wahrnehmbar ist. Um die Abweichungen der Prognosen zum Soll-Wert zu bestimmen, werden sogenannte Verlustfunktionen (engl. *Loss Function*) definiert. Bekannte Verlustfunktionen sind für Regressionsprobleme die mittlere quadratische Abweichung (engl. Mean Square Error)

$$L_{\text{MSE}} = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2 \quad (2.5)$$

oder für Klassifikationsprobleme die Kreuzentropie (engl. Cross Entropy)

$$L_{\text{CE}} = - \sum_{i=1}^N y_i \log \hat{y}_i \quad (2.6)$$

Es werden die Anzahl der vorhandenen Datensätze  $N$ , der zu erwartende Soll-Wert  $y_i$  und der durch das Netzwerk prognostizierte Wert  $\hat{y}_i$  verwendet (Sutton & Barto 2018).

Maschinelle Lernverfahren werden in drei Kategorien, die überwachten, die unüberwachten und die bestärkenden Lernverfahren, eingeteilt. Diese Verfahrenstypen werden in unterschiedlichen Aufgabenstellungen eingesetzt. Zwei dieser drei Verfahrensklassen werden im Folgenden detailliert.

## 2.2.2 Überwachtes Lernen

Überwachte Lernverfahren zeichnen sich dadurch aus, dass vor dem Training des neuronalen Netzwerks annotierte Trainingsdatenbeispiele generiert oder aufgezeichnet werden (Mohri et al. 2018). Eine Annotierung bedeutet, dass den Eingangsdaten bereits der korrekte Ausgangswert zugewiesen wird. Die überwachten Lernverfahren finden vermehrt in der Bild- und Signalverarbeitung Einzug. Zwei wesentliche Problemstellungen, die Regression und die Klassifikation, können durch überwachte Lernverfahren gelöst werden (Frochte 2021; Neapolitan & Jiang 2018). Bei der Regression muss das Ziel aus einem kontinuierlichen Bereich gefunden werden, während bei der Klassifikation der Eingabewert einer diskreten Zielmenge zugeordnet wird. Die Annotierung der Trainingsdaten kann unter Umständen sehr zeitintensiv sein. Häufig werden reale Datenbeispiele auch manuell annotiert. Um das Training eines neuronalen Netzwerks im überwachten Lernverfahren auf Erfolg zu überprüfen, werden von dem Trainingsdatensatz Datenbeispiele für einen Test- und Validierungsdatensatz ausgeschleust. Die Datenbeispiele des Test- und Validierungsdatensatzes sind damit nicht im Trainingsdatensatz enthalten.

### 2.2.3 Bestärkendes Lernen

Anders als überwachte Lernverfahren, werden bestärkende Lernverfahren nicht mit einem vorab generierten Datensatz trainiert. Das Training wird von einem Agenten gesteuert, der mit der Umwelt interagiert und ein künstliches, neuronales Netzwerk trainieren kann. Die Interaktion des Agenten wird durch Aktionen (engl. Actions), Zustände (engl. States) und Belohnungen (engl. Rewards) beeinflusst. Zuerst wählt der Agent eine Aktion aus, welche ausgeführt wird. Durch diese ausgeführte Aktion ergibt sich ein Zustand, der dem Agenten als Eingangswert zur Verfügung gestellt wird. Aus der gewählten Aktion und dem resultierenden Zustand wird eine Belohnung oder Bestrafung an den Agenten übergeben, welche beinhaltet, wie zielführend die ausgewählte Aktion war. Der Agent verfolgt das Ziel, Aktionen auszuwählen, um die erhaltene Belohnung zu maximieren. Das bestärkende Lernverfahren basiert somit auf dem Versuch und Irrtum Prinzip (engl. Trial-and-Error) (Frochte 2021).

Um die Interaktion des Agenten mit seiner Umwelt durch Aktionen, Zustände und Belohnungen mathematisch zu beschreiben, werden Markov Entscheidungsprozesse (engl. Markov Decision Processes) eingesetzt (Sutton & Barto 2018). Ein Markov Entscheidungsprozess wird durch ein Tupel mit folgenden Inhalten beschrieben:

- Menge an Zuständen  $S$
- Menge an Aktionen  $A$
- Transitionsmatrix  $P_{s,s'}^a$
- Belohnungsfunktion  $R(s)$
- Diskontierungsfaktor  $\gamma$

Die Transitionsmatrix

$$P_{s,s'}^a = \mathbb{P}[S_{t+1} = s' | S_t = s, A_t = a] \quad (2.7)$$

beschreibt die Wahrscheinlichkeit des Übergangs von einem Zustand  $s$  in einen Folgezustand  $s'$  durch eine ausgewählte Aktion  $a$ . Die Belohnungsfunktion

$$R(s) = \mathbb{E}[R_{t+1} | S_t = s, A_t = a] \quad (2.8)$$

belohnt anhand des Zustandes  $s$  die ausgewählte Aktion  $a$ . Das Ziel eines Markov Entscheidungsprozesses ist eine deterministische Strategie (engl. Policy)

$$\pi^* = S \rightarrow A \quad (2.9)$$

zu bestimmen. Der Einsatz von stochastischen Strategien und Explorationsmethoden dient dabei als Hilfsmittel, um diesen Prozess zu unterstützen und möglicherweise bessere Strategien zu entdecken, die auf einen deterministischen Zustand abzielen.

Während die Strategie  $\pi$  eine Aktion für einen Zustand  $s$  auswählt, würde die Strategie  $\pi^*$  durch die Aktionsauswahl den erwarteten Nutzen maximieren. Der erwartete Nutzen eines Zustand-Aktions-Paares wird auch als q-Zustand bezeichnet und kann als  $Q(s,a)$  beschrieben werden. Für einen Markov Entscheidungsprozess ergibt sich somit eine optimale Strategie

$$\pi^*(s) = \operatorname{argmax}_a [Q(s,a \in A)] = \operatorname{argmax}_\pi \mathbb{E} \left[ \sum_{t=0}^{\infty} R(s_t) | \pi \right] \quad (2.10)$$

Bestärkende Lernverfahren kommen dann zum Einsatz, wenn die Übergangswahrscheinlichkeit oder die Belohnungsfunktion unbekannt sind. Neben Atari-Spielen (Mnih et al. 2013) und Poker (Moravčík et al. 2017; Brown & Sandholm 2017) werden auch industrielle Aufgaben in der Robotik (Levine et al. 2016; Gandhi et al. 2017; Pinto et al. 2017) und dem autonomen Fahren (Pan et al. 2017) sowie Aufgaben in der Finanzbranche (Deng et al. 2016) mit bestärkenden Lernverfahren gelöst (François-Lavet et al. 2018).

### 2.2.4 Gefaltete, neuronale Netzwerke

Gefaltete, neuronale Netzwerke (engl. Convolutional Neural Network, kurz CNN) haben gegenüber vollständig verbundenen Netzwerken den Vorteil, Eingangsdaten in Matrixform direkt verarbeiten zu können. Dies stellt insbesondere in der Bildverarbeitung einen Vorteil dar. Durch die Einführung eines induktiven Bias können wesentliche Merkmale in Bilddaten durch Faltungen ermittelt werden. Diese Netzwerke wurden erstmals von LeCun (1989) vorgestellt und finden neben der Bildverarbeitung Verwendung in der Zeitreihenvorhersage, dem bestärkenden Lernen und der Spracherkennung (Krizhevsky et al. 2017; Simonyan & Zisserman 2014; He et al. 2016, van Oord et al. 2016; Zhang et al. 2019; Yin et al. 2017; Rusu et al. 2016; Dong et al. 2020).

Während in vollständig verbundenen Netzwerken die Matrixmultiplikation zwischen den Schichten angewendet wird, nutzt ein gefaltetes, neuronales Netzwerk den mathematischen Operator der Faltung (Goodfellow et al. 2016). Die Gewichtung der Eingangswerte und deren Integration durch den Faltungsoperator wird mittels

$$s(t) = \int_{-\infty}^{\infty} x(a)\omega(t-a)da = (x * \omega)(t) \quad (2.11)$$

berechnet. Die zeitabhängigen Eingangswerte  $x(t)$  liegen meist in Form von Sensordaten vor. Als Filter (engl. Kernel) wird der zeitabhängige Gewichtungsfaktor  $\omega(a)$  bezeichnet. Aus der Faltung gehen die Daten  $s$  hervor, welche auch als gefilterte Merkmalskarten (engl. Feature Maps) bezeichnet werden. Der Faltungsoperator wird in seiner Kurzform auch mit dem \*-Symbol dargestellt.

Für eine numerische Berechnung des Integrals wird die Zeitkomponente  $t$  mit ganzzahligen Werten diskretisiert. Die numerische Beschreibung des Faltungsoperators wird mittels

$$s(t) = (x * \omega)(t) = \sum_{a=-\infty}^{\infty} x(a)\omega(t-a) \quad (2.12)$$

berechnet.



Für zweidimensionale Eingabewerte, wie sie beispielsweise durch ein Bild in Matrixform  $I$  vorkommen, gilt

$$S(i,j) = (I * F)(i,j) = \sum_m \sum_n I(i + m \cdot b, j + n \cdot b) F(m,n) \quad (2.13)$$

welche die Filtermatrix  $F$  enthält. Das Filter wird bei der Faltung mit einer definierten Schrittweite (engl. Stride)  $b$  entlang der Bildelemente verschoben.

Neben einem Faltungsoperator werden häufig auch eine nichtlineare Aktivierungsfunktion sowie eine Bündelungsfunktion (engl. Pooling) beim Übergang zwischen den Faltungsschichten eingesetzt. Diese Bündelungsfunktion fasst umliegende Werte statistisch zu einem Wert zusammen und reduziert somit die Dimension der Eingabewerte und den Rechenaufwand. Die zwei gängigsten Bündelungsfunktionen sind:

- Maximal-Bündelung: der Maximalwert der Bildelemente wird ausgewählt (Zhou & Chellappa 1988)
- Durchschnittsbündelung: der Durchschnittswert der Bildelemente wird gebildet (Goodfellow et al. 2016, Zhou & Chellappa 1988)

### 2.2.5 Statistische Analyse einer binären Klassifizierung

Für die Bewertung der Leistungsfähigkeit eines binären Klassifikators, welcher in dieser Arbeit verwendet wird, wird eine Konfusionsmatrix (oder Wahrheitsmatrix, engl. *Confusion Matrix*) verwendet. Diese Matrix überprüft, wie viele der prognostizierten Vorhersagen richtigerweise oder fälschlicherweise einer Klasse zugeordnet wurden. Hierbei wird in folgende Felder der Matrix unterteilt:

- True Positive (TP): Die Prognose wurde richtigerweise als positiv klassifiziert.
- True Negative (TN): Die Prognose wurde richtigerweise als negativ klassifiziert.
- False Positive (FP): Die Prognose wurde fälschlicherweise als positiv klassifiziert.
- False Negative (FN): Die Prognose wurde fälschlicherweise als negativ klassifiziert.

Anhand der Einordnung der Prognosen relativ zu den tatsächlichen Werten können Kennzahlen aus der Konfusionsmatrix errechnet werden.

### Sensitivität

Die Sensitivität (engl. *Recall*)

$$\text{Sensitivität} = \frac{TP}{TP + FN} \quad (2.14)$$

wird auch als True-Positive-Rate bezeichnet und beschreibt den Anteil der korrekt positiv klassifizierten Datenbeispiele, im Verhältnis zur Gesamtheit der tatsächlich positiven Datenbeispiele.

### Präzision

Die Präzision (engl. *Precision*)

$$\text{Präzision} = \frac{TP}{TP + FP} \quad (2.15)$$

beschreibt den Anteil der korrekt klassifizierten Datenbeispiele im Verhältnis zu allen in dieser Kategorie klassifizierten Datenbeispielen.

### Genauigkeit

Die Genauigkeit (engl. *Accuracy*)

$$\text{Genauigkeit} = \frac{TP + TN}{TP + TN + FP + FN} \quad (2.16)$$

beschreibt die insgesamt richtig klassifizierten Datenbeispiele im Verhältnis zur Anzahl aller klassifizierten Daten.

### $F_1$ -Maß

Das  $F_1$ -Maß

$$F_1\text{-Maß} = \frac{2 \cdot \text{Sensitivität} \cdot \text{Präzision}}{\text{Sensitivität} + \text{Präzision}} \quad (2.17)$$

ist eine im maschinellen Lernen und der Datenanalyse weit verbreitete Metrik zur Bewertung der Leistungsfähigkeit eines Modells für Klassifizierungsaufgaben. Das Maß ist ein gewichtetes, harmonisches Mittel aus Präzision und Sensitivität, dessen Berechnung einen Wert im Intervall  $[0,1]$  liefert. Ein höherer Wert steht für ein leistungsstärkeres Modell. Der Einsatz des  $F_1$ -Maßes wird besonders in Situationen empfohlen, in denen ein Ungleichgewicht der Anzahl positiver oder negativer Datenbeispiele besteht.

## 2.3 Komponenten einer Griff-in-die-Kiste-Vereinzelungslösung

In diesem Abschnitt werden Komponenten einer Griff-in-die-Kiste-Vereinzelungslösung vorgestellt. Zu Beginn wird auf 3D-Bildsensoren und die Weiterverarbeitung der Tiefeninformationen in einer Objektlageerkennung eingegangen. Ferner werden Industrieroboter und deren Kinematiken vorgestellt. Um einem Industrieroboter das Greifen von Werkstücken zu ermöglichen, werden Greifsysteme und Greifplanungsalgorithmen verwendet und nachfolgend erläutert. Am Ende dieses Abschnitts werden weitere Roboterzellenperipherien und -komponenten, wie Kisten, Ablagen und zusätzliche Bildverarbeitung vorgestellt.

### 2.3.1 3D-Bildsensor und Objektlageerkennung

Bei einer Griff-in-die-Kiste-Vereinzelungslösung handelt es sich um einen bildverarbeitenden Prozess. Hierfür wird ein dreidimensionales Tiefenbild der Kistensituation erstellt und in einer Punktwolke weiterverarbeitet. Eine Kistensituation besteht hierbei aus dem Ladungsträger und den darin befindlichen Werkstücken. Um ein Tiefenbild erstellen zu können, wird ein 3D-Bildsensor verwendet. Ein 3D-Bildsensor wird nach bestimmten Kriterien ausgewählt, um für die jeweilige Griff-in-die-Kiste-Anwendung bestmögliche Tiefenbilder zu erzeugen. Ein Kriterium ist dabei das Messvolumen, welches ausreichend dimensioniert sein soll, um die gesamte Kistensituation aufzeichnen zu können. Zudem muss die Auflösung des 3D-Bildsensors ausreichen, um Werkstücke hinreichend genau lokalisieren zu können. Sensorhersteller unterscheiden hierbei zwischen lateraler Auflösung, welche sich auf die Bildebene bezieht und einer Höhenauflösung, welche orthogonal zur

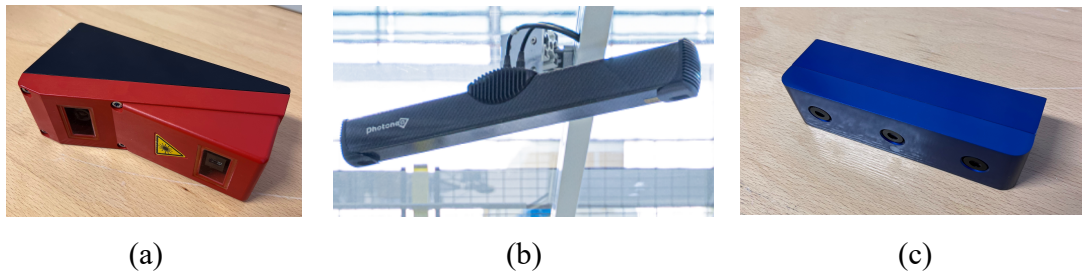


Abbildung 2.1: 3D-Bildsensoren: (a) Leuze LPS36, (b) Photoneo PhoXi 3D Scanner M, (c) Ensenso N20

lateralen Auflösung angegeben wird. Sensoren müssen in Abhängigkeit einer stationären oder robotergeführten Montage bestimmte Baugrößen einhalten und sollten robust für industrielle Produktionsumgebungen ausgelegt sein. Die 3D-Bildsensoren arbeiten mit unterschiedlichen Messverfahren. Neben Lichtlaufzeitmessung und Lasertriangulation kommen Streifenlichtprojektion und Stereokameras zum Einsatz.

### Lasertriangulation

Das Lasertriangulationsverfahren benötigt eine Laserquelle und eine oder mehrere Kameras. Die Laserquelle projiziert eine Linie oder auch nur einen Punkt auf die aufzunehmende Situation. Diese Linie oder dieser Punkt wird von der oder den Kameras aufgezeichnet, welche in einem bestimmten Abstand zur Laserquelle angeordnet sind. Anhand der Triangulation kann die Entfernung des Sensors zur Projektion bestimmt werden. Einen Sensor mit diesem Messverfahren stellt beispielsweise der Leuze LPS36 dar und ist in Abbildung 2.1 (a) dargestellt.

### Lichtlaufzeitmessung

Für die Lichtlaufzeitmessung wird ebenfalls eine Lichtquelle und eine Kamera benötigt. Hierbei wird durch Zeiterfassung die Differenz zwischen dem Entsenden und dem Empfangen des Lichtstrahls gemessen und daraus die Entfernung des Punktes der Tiefensituation berechnet. *Time-of-Flight*-Kameras nutzen dieses Verfahren.

### Streifenlichtprojektion

Bei der Streifenlichtprojektion wird ein Streifenmuster durch eine LED- oder Laserlichtquelle auf die Szene projiziert und mit einer oder mehreren Kameras aufgezeichnet. Die

Berechnung der Tiefeninformation geschieht in Anlehnung an das Triangulationsverfahren. Der in Abbildung 2.1 (b) dargestellte 3D-Bildsensor Photoneo PhoXi 3D Scanner M arbeitet mit diesem Verfahren.

### **Stereokamera**

Eine Stereokamera arbeitet mit zwei Kameras, die in einem bestimmten Abstand und Winkel zueinander angebracht sind. Beide Kameras zeichnen die Situation durch Graustufen- oder Farbbilder auf. Aufgrund des bekannten Abstands und Winkel der Kameras zueinander und dem Abstand von Merkmalen (Disparität) in den aufgezeichneten Bildern kann das Tiefenmaß durch Triangulation berechnet werden. In Abbildung 2.1 (c) ist ein Ensenso N20 dargestellt, der nach dem erläuterten Prinzip arbeitet.

Die Messverfahren der unterschiedlichen 3D-Sensoren haben Vor- und Nachteile im Einsatz in einer Griff-in-die-Kiste-Applikation. Während Stereokameras in der Auflösung und Genauigkeit starke Performanzen aufweisen, sind sie anfällig gegenüber Störfaktoren. Lichtlaufzeitmessverfahren schneiden in der Geschwindigkeit der Bildaufnahme sehr gut ab, während die Auflösung und Genauigkeit durch andere Messverfahren deutlich besser erreicht werden. Die Lasertriangulation und die Streifenlichtprojektion weisen Stärken in der Auflösung und Genauigkeit auf, während sie kleine Schwächen in der Geschwindigkeit der Bildaufnahme, der Robustheit und der Sichtbarkeit gegenüber Abschattungen aufweisen. Der Einsatz der Messverfahren zur Aufnahme der Tiefeninformation muss anhand der genannten Vor- und Nachteile für bestimmte Umgebungs- und Werkstückbeschaffenheiten evaluiert werden. In dieser Arbeit wird das Streifenlichtprojektionsverfahren durch einen Photoneo PhoXi 3D Scanner M in der realen Roboterzelle eingesetzt.

Nachdem das Tiefenbild in Form einer Punktwolke aufgezeichnet wurde, wird es in einem Objektlageerkennungsverfahren weiterverarbeitet. Zur Objektlageerkennung bestehen unterschiedliche Verfahren. Neben maschinellen Lernverfahren, die in einem Raster, welches über das Tiefenbild gelegt wird, Werkstücklagen prognostizieren (Kleeberger & Huber 2020), werden analytische, objekt- oder ansichtenbasierte Verfahren eingesetzt (Palzkill 2014). Während objektbasierte Objektlageerkennungsverfahren mit vollständigen

Beschreibungen der Werkstückgeometrie arbeiten und damit anhand von Kanten, Kurven, Oberflächen und ganzen Volumen Werkstücke lokalisieren, arbeiten ansichtenbasierte Verfahren mit Merkmalen und Strukturbeschreibungen aus vorher erstellten Ansichten von Werkstückgeometrien. Hierfür erstellt Palzkill (2014) Datenbanken mit Ansichten der Werkstückgeometrien und sucht nach den generierten Ansichten der Werkstücke in den Tiefeninformationen. Das ansichtenbasierte Objektlageerkennungsverfahren von Palzkill (2014) wird in dieser Arbeit verwendet.

### 2.3.2 Industrieroboter

Industrieroboter werden für unterschiedliche Aufgaben in Produktionsanlagen eingesetzt. Neben Schweiß-, Lackier- und Montageaufgaben übernehmen die Roboter repetitive Prozesse in Handhabungsprozessen. Laut ISO 8373 wird ein Industrieroboter als programmierbarer Manipulator beschrieben, der drei oder mehr Achsen besitzt und für industrielle Prozesse eingesetzt wird.

Verschiedene Aspekte werden beachtet, um einen geeigneten Industrieroboter für eine Griff-in-die-Kiste-Applikation auszuwählen. Der Arbeitsraum des Roboters sollte dabei ausreichend dimensioniert werden, um Werkstücke in sämtlichen Orientierungen aus der Kiste greifen und diese in bestimmten Lagen in eine Ablage befördern zu können. Zudem sollte die Genauigkeit des Roboters ausreichen, um vorgegebene Posen für einen erfolgreichen Greifprozess anzufahren. Hierbei sind sowohl die Wiederholgenauigkeit als auch die Absolutgenauigkeit ein Maß zur Auswahl. Ferner wird eine ausreichende Anzahl an möglichen Freiheitsgraden der Handhabungskinematik benötigt, um die Zugänglichkeit zu beliebigen Werkstücklagen in der Kiste zu erreichen. Hierfür muss die Handhabungskinematik um jede vorhandene Roboterachse eine rotatorische Drehung ermöglichen und in jeder Raumdimension verfahrbar sein (Czichos 2015). Die Traglast der Handhabungskinematik muss so ausgelegt werden, dass das Gewicht eines Greifsystems und eines Werkstücks diese nicht überschreiten. Um eine Griff-in-die-Kiste-Anwendung möglichst effizient auszulegen, sollte die Verfahrensgeschwindigkeit und -beschleunigung des Roboterendeffektors ausreichend dimensioniert werden. Für Handhabungsaufgaben werden unterschiedliche Kinematiken eingesetzt. Diese werden in serielle Kinematiken und Parallelkinematiken unterschieden.

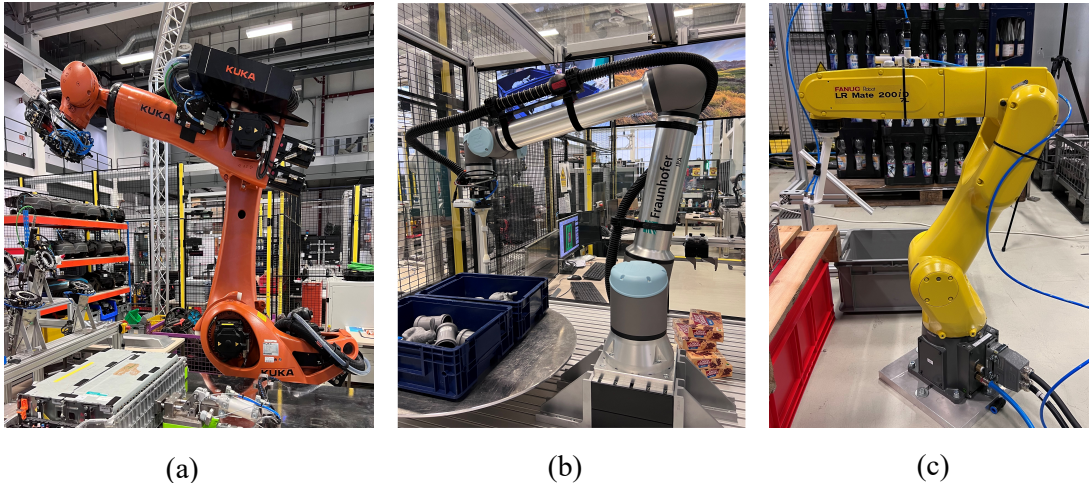


Abbildung 2.2: Industrieroboter: (a) KUKA KR100, (b) Universal Robot UR10e, (c) Fanuc LR Mate 200iD-7L

### Serielle Kinematik

Serielle Kinematiken sind so aufgebaut, dass nur eine kinematische Kette entlang des Manipulators verfolgt wird und rotatorische sowie translatorische Achsen nacheinander entlang der Kette angeordnet sind. Typischerweise werden vier bis sieben Achsen vorgesehen, welche alle von Aktoren bewegt werden können. Sechssachsige, serielle Kinematiken sind häufig eingesetzte Handhabungskinematiken für automatische Zuführsysteme. In Abbildung 2.2 sind drei sechssachsige, serielle Kinematiken unterschiedlicher Hersteller abgebildet.

### Parallelkinematik

Eine Parallelkinematik unterscheidet sich im Vergleich zu einer seriellen Kinematik durch multiple Verbindungen des Endeffektors durch mehrere kinematische Ketten. Trotz der hohen Verfahrensgeschwindigkeit, die durch Parallelkinematiken erreicht werden kann, sind diese durch eine erhöhte Kollisionsgefahr der kinematischen Ketten mit der Roboterzellenperipherie den seriellen Kinematiken unterlegen. Serielle Kinematiken haben einen deutlich größeren Arbeitsraum als Parallelkinematiken, sind in ihrer Dynamik allerdings stärker limitiert (Pott & Dietz 2019). Parallelkinematiken hingegen weisen eine deutlich höhere Wiederhol- und Absolutgenauigkeit auf.

Die Berechnung der Kinematik wird in Vorwärts- und Rückwärtstransformation unterschieden. Die Vorwärtstransformation berechnet aus den Achswinkeln der Roboterkinematik

die Pose des Endeffektors in Bezug auf das Basiskoordinatensystem, während die Rückwärtstransformation, auch inverse Kinematik genannt, aus der Pose des Endeffektors die Achswinkel berechnet. Es existieren hierfür diverse Lösungsverfahren (Denavit & Hartenberg 1955; Hartenberg & Denavit 1964) und für manche kinematischen Ketten explizite Berechnungen (Heiß 1985; Heiß 1986). Zur Erleichterung der Bestimmung der inversen Kinematik wird in der Robotik oft konstruktiv ein Handwurzelgelenk eingeführt. Dieses beschreibt den Schnittpunkt der vierten bis sechsten Achse (Paul & Zhang 1986).

Um einen Industrieroboter von Pose zu Pose bewegen zu können, müssen Bewegungsprofile angelegt werden. Unterschieden wird hierbei zwischen Linearbewegungen (LIN) und Point-to-Point-Bewegungen (PTP). Zu Beginn der PTP-Bewegung werden alle Achswinkeldifferenzen zur Ursprungsposition berechnet und die Achse mit der größten Differenz am schnellsten bewegt. Alle anderen Achsen fahren mit einer geringeren Geschwindigkeit, sodass die Zielpose von allen Achsen zum selben Zeitpunkt erreicht wird. Bei fest programmierten Bewegungspfaden des Roboters ist diese Bewegungsart am besten geeignet, da mit dieser die maximale Geschwindigkeit des Roboters erreicht werden kann. LIN-Bewegungen hingegen verfahren von Pose zu Pose entlang einer Geraden. Für Bewegungen in engen Räumlichkeiten, beispielsweise in eine Kiste hinein, bietet sich diese Bewegungsart an. Die maximale Verfahrensgeschwindigkeit des Roboters kann hierbei meist nicht erreicht werden.

### 2.3.3 Greifsysteme

Damit ein Industrieroboter in einer Griff-in-die-Kiste-Anwendung ein Werkstück greifen kann, benötigt dieser ein Greifsystem. Dieses System wird an den Flansch des Roboterarms angebracht. Zur Auswahl eines Greifsystems müssen verschiedene Aspekte beachtet werden. Zum einen sollte ein Greifsystem möglichst kompakt konzipiert werden, damit die Störkontur beim Hineinfahren in eine Kiste keine Kollision verursacht. Zum anderen sollte das Greifsystem in der Lage sein, möglichst viele unterschiedliche Greifposen an verschiedenen Werkstücklagen in der Kiste auszuführen. In Abhängigkeit der Werkstückgeometrie und -beschaffenheit können unterschiedliche Greifprinzipien zum Einsatz kommen. Neben Sauggreifern, welche mithilfe von Unterdruck oder Volumenstromevakuierung Werkstücke ansaugen können, existieren Magnetgreifer, die die magnetischen Eigenschaften der Werk-



stücke ausnutzen. Ferner werden Klemmgreifer eingesetzt. Hierbei werden Greiferbacken an einem Greifsystem angebracht, die durch rotatorische oder translatorische Bewegungen Werkstücke in den Greiferbacken einschließen. Greifsysteme werden in zwei Kategorien eingeteilt.

### **Formschluss**

Ein Formschluss wird erzeugt, indem der Greifer auf die Werkstückgeometrie geometrisch angepasst wird, so dass er eine Kavität besitzt, in welche das Werkstück aufgenommen werden kann. Beispielsweise werden bei Greiferbacken Aussparungen, Fasen und Rundungen vorgesehen, welche beim Zusammenfahren der Greiferbacken ohne zusätzliche Greifkraft einen stabilen Griff des Werkstücks erzeugen. Es existieren zudem pneumatische Greifer, die mit Formschluss Werkstücke handhaben können. Ein Beispiel hierfür ist der *FlexShapeGripper* der Firma Festo Vertrieb GmbH & Co. KG.

### **Kraftschluss**

Verschiedene Greifprinzipien arbeiten mithilfe des Kraftschlusses. Neben Klemmgreifern, die durch das Zusammen- oder Auseinanderfahren der Greiferbacken Werkstücke einklemmen können und damit eine Klemmkraft aufbringen, bestehen weitere Greiftypen dieses Prinzips, wie beispielsweise Sauggreifer und Magnetgreifer. Ein Sauggreifer besitzt meist einen oder mehrere Saugnäpfe, welche durch erzeugten Unterdruck eine Haltekraft auf das Werkstück ausüben. Gleiches geschieht bei Magnetgreifern. Durch die Bestromung einer elektrischen Spule ist dieser in der Lage, auf metallisch-magnetische Werkstücke eine Haltekraft auszuüben. Weitere Greifprinzipien arbeiten mit dem Erzeugen von Adhäsionskräften.

## **2.3.4 Greifplanung**

Ein Industrieroboter mit Greifsystem benötigt in einer Griff-in-die-Kiste-Applikation eine Greifplanung, welche anhand der lokalisierten Werkstücklagen Greifposen sucht und eine Entnahmepfad des Roboters plant. Bevor Greifposen an den Werkstücken in der Kiste gesucht werden können, müssen diese definiert werden. Dieser Vorgang kann manuell oder automatisch erfolgen.

### Greifposendefinition

Die manuelle Greifposendeklaration kann in einer Griff-in-die-Kiste-Software erfolgen. In dieser Software können unter Berücksichtigung des CAD-Modells des Werkstücks und des Greifsystems Posen dieser zwei Geometrien zueinander definiert werden, welche eine geeignete Greifpose ergeben. Es besteht zudem die Möglichkeit, für jede Greifpose translatorische und rotatorische Freiheitsgrade zu definieren. Die möglichen Greifposen werden in Transformationsmatrizen mit definierten Freiheitsgraden  $f$  gespeichert. Eine automatisierte Greifposengenerierung (Khalid et al. 2021; Kleeberger et al. 2021) kann in einer Simulationsumgebung stattfinden. In dieser Umgebung können Greifer und Werkstück automatisch, relativ zueinander bewegt und sowohl für Saug- und Magnetgreifer als auch für Klemmgreifer mögliche Greifposen definiert werden. Sollte das zu greifende Werkstück Bereiche enthalten, welche für Griffe ungeeignet oder nicht erwünscht sind, so ist im Nachgang zusätzlicher Input eines Experten notwendig.

### Heuristische Greifposensuche

Nachdem Greifposen definiert wurden, gilt es in einem Griff-in-die-Kiste-Prozess schnell und effektiv mögliche Greifposen für die gefundenen Werkstücklagen zu finden. Eine Möglichkeit, dies zu erreichen, besteht in einer heuristischen Greifposensuche. Eine Heuristik ist eine Methode, mit begrenztem Wissen und Zeit zu einer wahrscheinlichen Aussage und praktikablen Lösung zu gelangen. Eine heuristische Greifposensuche wurde von Spenrath (2022) vorgestellt. Dieser Suche liegt ein Suchbaum zu Grunde, der an einem Wurzelknoten gestartet wird. Für jedes lokalisierte Werkstück wird ein Kindknoten erzeugt, welcher durch Greifposen mit Freiheitsgraden, zusätzliche Achsen im Greifer und Bewegungspfadposen des Roboters expandiert werden kann. Um die bestmöglichen Knoten zu expandieren, wird eine Funktion

$$h(x) = c_s \cdot (1 - w_d) + c_d \cdot w_d \quad (2.18)$$

eingeführt, welche von verschiedenen statischen  $c_s$  und dynamischen Kostenfaktoren  $c_d$  sowie einem Gewichtungsfaktor  $w_d$  abhängt. Die dynamischen Kostenfaktoren können sich im Laufe der Greifposensuche verändern und vermeiden die Konzentration der Suche

auf einzelne Knoten. Beispiele für die statischen Kostenfaktoren sind die Höhe der Werkstücklage im Ladungsträger im Vergleich zu allen anderen lokalisierten Werkstücken, der Konfidenzwert der Objektlageerkennung des Werkstücks oder die Verkippung des Greifers, welche benötigt wird, um die Greifpose zu erreichen.

Wenn eine Greifpose für die Werkstücklage in der Kistensituation gefunden wurde, wird diese Pose an die Robotersteuerung übergeben. Es ist allerdings notwendig, zusätzliche Pfadposen vor und nach der Greifpose in den Entnahmepfad einzufügen, um mögliche Kollisionen zu vermeiden, die in direkten Roboterpfaden beispielsweise von Ablageposen zum Werkstück in der Kiste auftreten können.

### **Entnahmepfadplanung**

Die Entnahmepfadplanung plant die Roboterbahn von einer bekannten Roboterpose über der Kiste hin zu einer berechneten Greifpose (Spenrath & Pott 2017). Hierfür wird zunächst ein *Interface-Point* definiert. Der *Interface-Point* ist eine fest programmierte Roboterpose in der Mitte über der Kiste, die für jede Werkstücklage gleich ist und wird als Startpose des Entnahmepfads festgelegt. In Abbildung 2.3 (a) ist diese Pose visualisiert. Der Pfad zu dieser Pose kann mit PTP-Bewegungen umgesetzt werden. Die erste, auf die Werkstücklage angepasste Roboterpose, befindet sich in einem vorab definierten Abstand in der  $z$ -Koordinate des Kistenkoordinatensystems über dem zugreifenden Werkstück. Die Orientierung des Greifers ist hierbei bereits auf die berechnete Greifpose angepasst. Zusätzlich kann die maximale Abweichung des Werkzeugmittelpunkts, auch Tool Center Point (TCP) genannt, zur Kistenmitte angegeben werden. Die beschriebene Pose wird als *Before-Object-Point* deklariert und ist in Abbildung 2.3 (b) visualisiert. Zur kollisionsfreien Bewegung zu dieser Pose wird eine LIN-Bewegung verwendet. Nach dieser Pose wird die Greifpose am Werkstück mit einer LIN-Bewegung angefahren und der Griff ausgeführt. Die Greifpose am Werkstück in der Kiste ist in Abbildung 2.3 (c) dargestellt. Abbildung 2.3 (d) zeigt zusätzliche Pfadposen nach der Greifpose, welche beispielsweise für eine Enthakung bestimmt werden können. Diese Posen sind optional und werden im Laufe dieser Arbeit entwickelt. Nach den optionalen Enthakungspfadposen werden der *Before-Object-Point* (Abbildung 2.3 (e)) und der *Interface-Point* (Abbildung 2.3 (f)) durch eine LIN-Bewegung

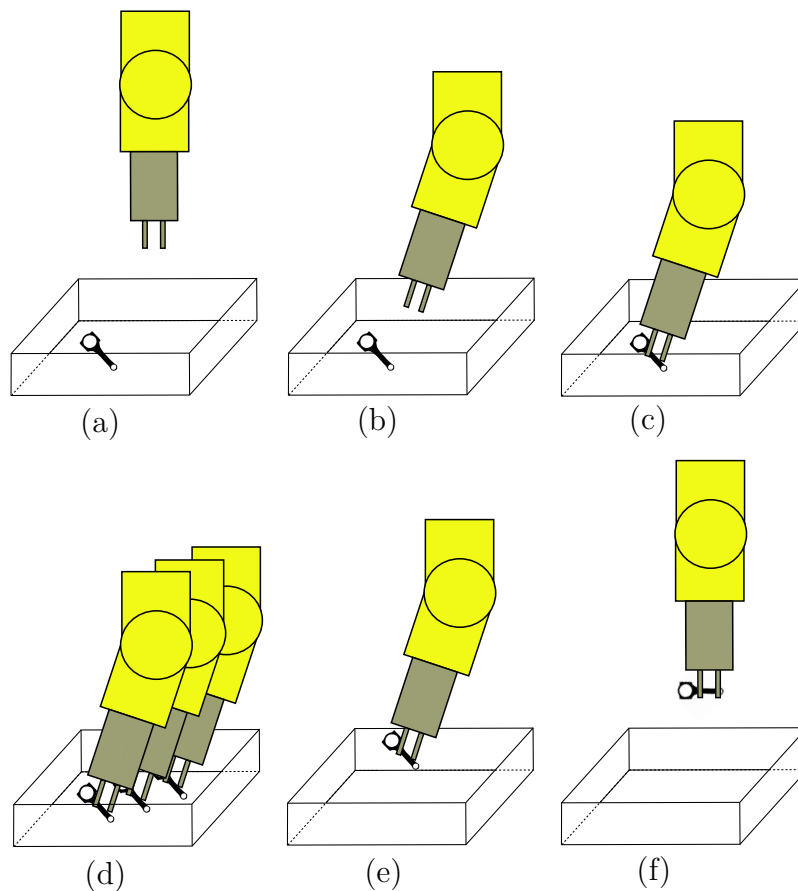


Abbildung 2.3: Entnahmepfadposen am Beispiel eines Klemmgreifers bei der Vereinzelung von Pleueln: (a) *Interface-Point*, (b) *Before-Object-Point*, (c) Greifpose, (d) optionale zusätzliche Pfadposen zum Beispiel für eine Enthakung, (e) *Before-Object-Point*, (f) *Interface-Point*.

abgefahren und der Greifprozess in der Kiste beendet. Weitere Pfadposen für Bewegungen des Roboters in der Roboterzelle oder Ablageposen der Werkstücke können gesondert programmiert oder deklariert werden.

### Kollisionserkennung

Kollisionen zwischen Roboter greifer, Roboter und Zellenperipherie gilt es aufgrund möglicher Beschädigungen an den Komponenten, fehlerhafter Vereinzelungen und Anlagenstopps zu vermeiden. Kollisionen können durch Kraft-Momenten-Sensorik (KMS) erkannt werden. Dies stellt sich bei höheren Geschwindigkeiten und Beschleunigungen des Roboterendeffektors als schwierig dar. Je größer die Kräfte und Momente einer erfolgreichen Vereinzelung eines schweren Werkstücks sind, desto robuster müssen die Grenzwerte für einen Abbruch

des Griffes durch den KMS deklariert werden. Dies kann bereits zu Beschädigungen und zum Stillstand führen. Eine deutlich sensiblere Lösung ist die Kollisionserkennung mit dem *Raytracing-Verfahren* (Glassner 1989). Dieses Verfahren nutzt Strahlenverfolgung, welche im Fall einer Griff-in-die-Kiste-Anwendung durch einen 3D-Bildsensor ausgestrahlt werden. Dieses Verfahren ist hauptsächlich bei Sensoren möglich, welche mit der Methode der Lichtlaufzeitmessung arbeiten. Tiefenbilder von Sensoren, die mit den Methoden der Lasertriangulation oder von Stereokameras arbeiten können jedoch in ausreichendem Maße angenähert werden, sodass das *Raytracing-Verfahren* auch hierfür angewendet werden kann. Die Strahlen werden durch die Messpunkte der Punktwolke in zwei Bereiche unterteilt. Der Bereich zwischen dem Sensor und der Punktwolke wird hierbei als kollisionsfreier Bereich deklariert. In diesem Bereich ist keine Kollision zu erwarten und mögliche Pfadposen können angefahren werden. Der Bereich aus Sensorperspektive hinter den Messpunkten der Punktwolke gilt als unsicherer Bereich. Hier ist keine Kenntnis von Kollisionsgegenständen vorhanden. Pfadposen sollten in diesem Bereich vermieden werden. Aufgrund des Rauschverhaltens der Sensoren und mit Toleranz gemessenen Punkten der Punktwolke können Situationen abgelehnt werden, in welchen nicht mit Gegenständen in der aufgezeichneten Situation sondern durch Ungenauigkeiten mit Messpunkten Kollisionen auftreten können. Diese Situationen können durch Verwendung von Grenzwerten abgefangen werden. Beispielsweise wird für eine bestimmte Geometrie eine maximale Anzahl an Kollisionen mit Messpunkten der Punktwolke erlaubt. Mit diesem Verfahren (Spenrath et al. 2012) ist es möglich, alle Geometrien in der vorhandenen Situation auf Kollisionen zu prüfen. Hierunter fallen das Greifsystem, der Roboter, die Kiste, die Werkstücke in der Kiste, das gegriffene Werkstück und weitere Hindernisse in der Roboterzelle.

### 2.3.5 Roboterzellenperipherie

In einer Roboterzelle einer Griff-in-die-Kiste-Anwendung sind neben dem 3D-Bildsensor, dem Industrieroboter inklusive Steuerung und dem Greifsystem weitere Komponenten vorhanden. Beispielsweise werden übergeordnete Steuerungssysteme eingesetzt, die die Sicherheit der Roboterzelle und weitere Komponenten, wie beispielsweise weitere Roboter oder Maschinen, steuern.

Im Produktionsumfeld wird für die Vereinzelung von Werkstücken, die durch ein Griff-in-die-Kiste-System erfolgen kann, unterschiedliche Kisten eingesetzt. Neben unterschiedlichen Abmessungen besitzen die Kisten zumeist auch geometrisch unterschiedliche Kistenränder und können neben geschlossenen Kistenwänden auch Wände aus Gitternetzen besitzen. Die Auswahl der Kisten wird meist hinsichtlich der logistischen Prozesse und der Werkstückabmessungen getroffen.

Das Ziel einer Vereinzelung besteht darin, Werkstücke aus beliebiger Lage in einer Kiste geordnet in einen Werkstückträger, eine Ablagestation oder direkt in eine Maschine einzulegen. Diese Ablagen sind zumeist mit Pufferspeichern versehen, um Toleranzen in den Zykluszeiten der Vereinzelung auszugleichen. Die Ablagevorrichtungen können auch Zwischenablagen beinhalten, in welche die gegriffenen Werkstücke abgelegt, umorientiert und dann wieder gegriffen werden. Hierdurch können Ablagen bei asymmetrischen Werkstückgeometrien in jeden Lagen gewährleistet werden. Die Pfadplanung der Ablage der gegriffenen Werkstücke sowie deren Kollisionserkennung kann, wie bereits die Entnahmepfadplanung, durch eine Griff-in-die-Kiste-Software durchgeführt werden.

Um genauere Ablageergebnisse zu erzielen, wird oft eine zusätzliche zwei- oder dreidimensionale Bildverarbeitung eingesetzt. Durch diese Bildverarbeitungssysteme werden die gegriffenen Werkstücke im Greifer vor einen Bildsensor gehalten und die Lage der Werkstücke im Greifer genauer berechnet. Diese Methode bietet sich bei Werkstückgeometrien mit kleinen, geometrischen Merkmalen an, die nur schwer durch eine erste, dreidimensionale Objektlageerkennung lokalisiert werden können.

# **3 Entwicklung einer Methode zur Verhakungserkennung von Werkstücken**

In diesem Kapitel wird ein Verfahren entwickelt, um verhakte Situationen der Werkstücke in einer sortenrein gefüllten Kiste zu erkennen. Dieses Verfahren stellt die Basis für das im folgenden Kapitel entwickelte Enthakungsverfahren dar. In Abschnitt 3.1 wird zunächst auf die Ausgangslage vorhandener Daten eingegangen und zudem die Aufbereitung benötigter Daten beschrieben. Der Transfer der vorhandenen Daten von der Simulationsumgebung in die Realität wird in Abschnitt 3.2 beleuchtet. Das Verfahren verwendet eine künstliche neuronale Netzwerkarchitektur, welche in Abschnitt 3.3 dargestellt und deren Trainingsmethode vorgestellt wird. In Abschnitt 3.4 wird die Integration des Verfahrens in das Griff-in-die-Kiste-Gesamtsystem beschrieben und am Ende des Kapitels in Abschnitt 3.5 zusammengefasst.

## **3.1 Ausgangslage, Generierung und Aufbereitung benötigter Daten**

Um ein neuronales Netzwerk zu trainieren, werden Trainingsdaten benötigt. Zu Beginn dieses Abschnitts werden die Daten der Bildsensorik und der Objektlageerkennung vorgestellt, die notwendig sind, um eine Verhakung zu erkennen. Zu möglichen Daten gehören sowohl real aus einer Roboterzelle aufgenommene als auch synthetisch in Simulationsszenen generierte Situationen. Die synthetische Datengenerierung wird in diesem Abschnitt erläutert. Um die benötigte Zeit zur Datengenerierung zu reduzieren, wird eine Methode zur Datenaugmentierung vorgestellt.



Abbildung 3.1: Punktwolke einer realen Kistensituation in der Roboterzelle in orthogonaler Projektion. Die z-Koordinate eines Punkts ist verantwortlich für die Farbgebung in der Darstellung.

### 3.1.1 Daten der Bildsensorik und der Objektlageerkennung

In Abschnitt 2.3.1 wurde gezeigt, dass in einer Griff-in-die-Kiste Applikation zur Erfassung der Kistensituation dreidimensionale Bildsensorik eingesetzt wird. Diese Bildsensorik liefert eine Punktwolke, wie sie in Abbildung 3.1 dargestellt ist. Eine Punktwolke besteht aus Punkten, die durch ihre jeweiligen Raumkoordinaten der Tiefeninformation angeordnet sind. Um diese Punktwolke zweidimensional abzubilden, wird ein Graustufenbild erstellt, in welchem die Tiefenwerte durch die Grauwerte abgebildet werden. Je weiter ein Punkt beispielsweise vom Sensor entfernt liegt, desto weißer wird dessen Projektion abgebildet. Die Tiefeninformation wird dabei vom Kistenboden bis zum Kistenrand normalisiert. Das entstehende Graustufenbild wird auch Tiefenkarte genannt. Die Tiefenkarten dienen als Eingangswerte vieler maschineller Lernverfahren. Durch Objektlageerkennungsmethoden (Palzkill 2014) werden anhand der Punktwolke Werkstückposen der bekannten Werkstückgeometrien erkannt. Die Objektlageerkennung kann sowohl objektbasiert, ansichtenbasiert als auch mittels maschineller Lernmethoden erfolgen. Die Werkstückposen werden der Greifplanung sowie den Methoden zur Verhakungserkennung und Enthakung zur Verfügung gestellt. In Abschnitt 3.1.2 werden verschiedene Transformationsmatrizen zur Projektion in unterschiedliche Koordinatensysteme vorgestellt und weiterverarbeitet.



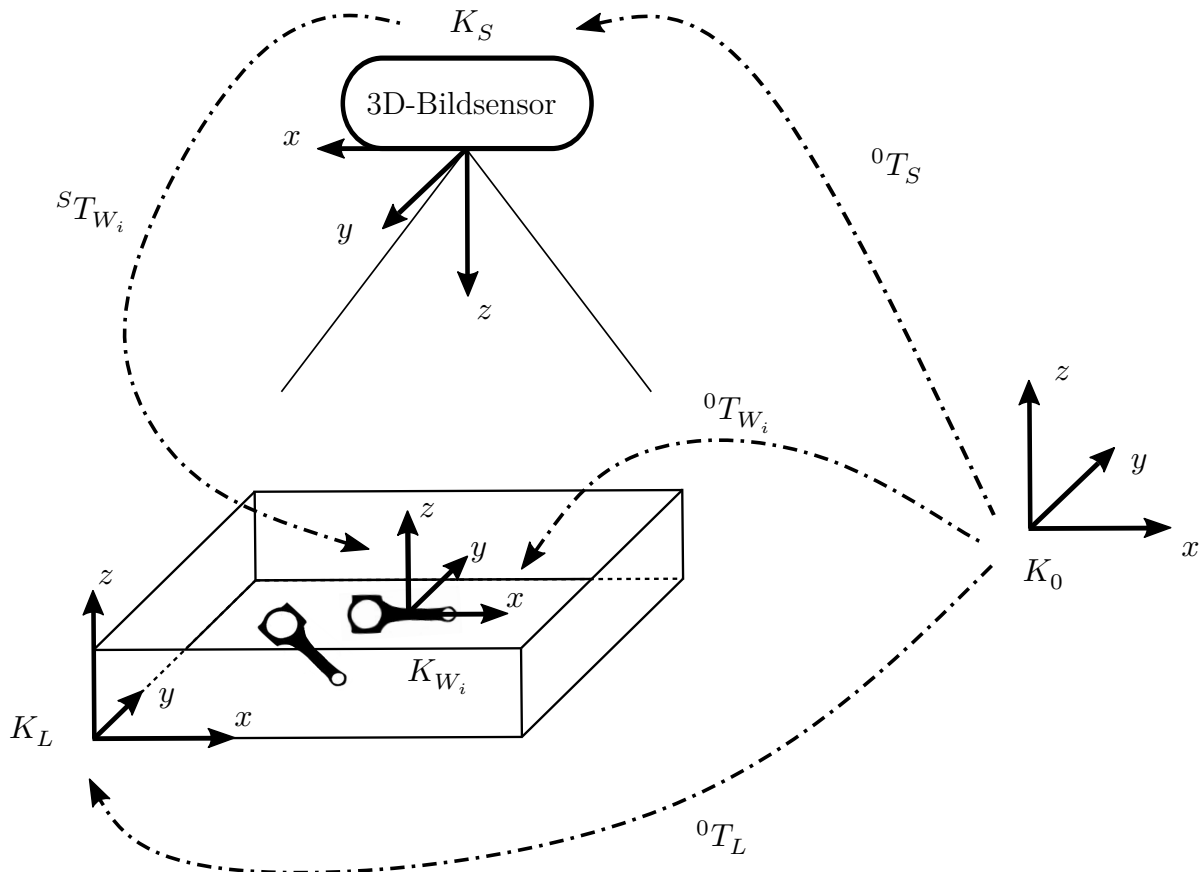


Abbildung 3.2: Übersicht von Koordinatensystemen  $K$  und Transformationsmatrizen  $T$

Neben dem Tiefenbild und den lokalisierten Werkstücklagen  ${}^0T_{W_i}$ , werden die lokalisierte Ladungsträgerlage  ${}^0T_L$  und die Sensorlage  ${}^0T_S$  relativ zum Weltkoordinatensystem in Transformationsmatrizen an die Verhakungserkennungs- und Enthakungsansätze sowie an die Greifplanung übergeben. Der Ladungsträger wird mittels eines vordefinierten CAD-Modells im Tiefenbild lokalisiert (Palzkill 2014). Die Sensorlage, relativ zum Weltkoordinatensystem, wird mit einer Hand-Auge-Kalibrierung (Spenrath et al. 2013) ermittelt. Die Roboterbasisposition wird im Ursprung des Weltkoordinatensystems  $K_0$  angenommen. Die Übersicht über alle Koordinatensysteme und Transformationsmatrizen, die für den Tiefenkartenzuschnitt benötigt werden, ist in Abbildung 3.2 dargestellt.

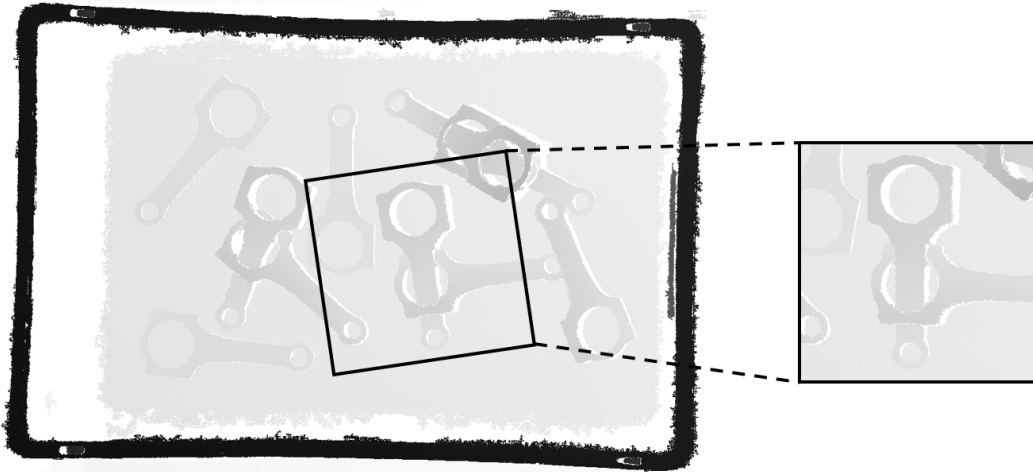


Abbildung 3.3: Beispiel eines Tiefenkartenzuschnitts (Moosmann et al. 2020)

### 3.1.2 Tiefenkartenzuschnitt

Um die Fokussierung des Trainings der maschinellen Lernmethode zur Verhakungserkennung auf das verhakete Werkstück und dessen direkte Nachbarschaft zu lenken, wird die Tiefenkarte der Kistensituation zugeschnitten. Ein beispielhafter Zuschnitt der Größe  $128 \times 128$  Pixel ist in Abbildung 3.3, rechts, dargestellt.

Die Zuschnitte werden entlang des Werkstückes  $W_i$  ausgerichtet. Der Mittelpunkt eines Zuschnittes ist der Ursprung des Werkstückkoordinatensystems  $K_{W_i}$ . Um den Zuschnitt zu erstellen, werden neben der gesamten Tiefenkarte der Kistensituation, Transformationsmatrizen

$${}^0T_X = \begin{pmatrix} \overrightarrow{0r_X^x} & \overrightarrow{0r_X^y} & \overrightarrow{0r_X^z} & \overrightarrow{0p_X} \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 0r_X^{x_1} & 0r_X^{y_1} & 0r_X^{z_1} & 0p_X^x \\ 0r_X^{x_2} & 0r_X^{y_2} & 0r_X^{z_2} & 0p_X^y \\ 0r_X^{x_3} & 0r_X^{y_3} & 0r_X^{z_3} & 0p_X^z \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (3.1)$$

benötigt. Die Transformationsmatrix des Werkstückes  ${}^0T_{W_i}$  relativ zum Weltkoordinatensystem  $K_0$  und die Transformationsmatrix des Ladungsträgers  ${}^0T_L$  werden für das Zuschneiden der Tiefenkarte verwendet.

Gesucht wird eine Transformationsmatrix  ${}^{C_i}T_S$ , welche den 3D-Bildsensor virtuell, relativ

zur gesamten Tiefenkarte, in ein neues Objektkoordinatensystem verschiebt und entlang der definierten Werkstückachse um  $z$  rotiert, damit der entsprechende Tiefenkartenzuschnitt zustande kommt und aufgezeichnet werden kann.

Die Transformationsmatrix vom Weltkoordinatensystem  $K_0$  zu der in die Kistenebene projizierten Werkstücklage wird als  ${}^0T_{C_i}$  deklariert. Diese unterscheidet sich in Abbildung 3.2 nicht von  ${}^0T_{W_i}$ , da dieses Werkstück bereits in der Kistenbodenebene liegt.

Da die  $x$ -Achse von  $K_{W_i}$  auch schief im Raum stehen kann und damit  $\overrightarrow{{}^0r_{W_i}^z} \nparallel \overrightarrow{{}^0r_L^z}$ , wird diese in die Kistenbodenebene projiziert. Hierfür wird zunächst das Kreuzprodukt aus der  $x$ -Achse von  $K_{W_i}$  ( $\overrightarrow{{}^0r_{W_i}^x}$ ) mit der  $z$ -Achse des Ladungsträgerkoordinatensystems  $K_L$  ( $\overrightarrow{{}^0r_L^z}$ ) mittels

$$\overrightarrow{{}^0r_{C_i}^y} = \overrightarrow{{}^0r_L^z} \times \overrightarrow{{}^0r_{W_i}^x} = \begin{pmatrix} 0r_L^{z1} \\ 0r_L^{z2} \\ 0r_L^{z3} \end{pmatrix} \times \begin{pmatrix} 0r_{W_i}^{x1} \\ 0r_{W_i}^{x2} \\ 0r_{W_i}^{x3} \end{pmatrix} = \begin{pmatrix} 0r_{C_i}^{y1} \\ 0r_{C_i}^{y2} \\ 0r_{C_i}^{y3} \end{pmatrix} \quad (3.2)$$

berechnet. Die dadurch entstehende orthogonale  $y$ -Achse  $\overrightarrow{{}^0r_{C_i}^y}$  in der Kistenbodenebene wird in einem folgenden Schritt mittels

$$\overrightarrow{{}^0r_{C_i}^x} = \overrightarrow{{}^0r_{C_i}^y} \times \overrightarrow{{}^0r_L^z} = \begin{pmatrix} 0r_{C_i}^{y1} \\ 0r_{C_i}^{y2} \\ 0r_{C_i}^{y3} \end{pmatrix} \times \begin{pmatrix} 0r_L^{z1} \\ 0r_L^{z2} \\ 0r_L^{z3} \end{pmatrix} = \begin{pmatrix} 0r_{C_i}^{x1} \\ 0r_{C_i}^{x2} \\ 0r_{C_i}^{x3} \end{pmatrix} \quad (3.3)$$

zur Projektion der  $x$ -Achse in einem Kreuzprodukt mit der  $z$ -Achse des Ladungsträgerkoordinatensystems  $K_L$  ( $\overrightarrow{{}^0r_L^z}$ ), ebenfalls parallel zur Kistenbodenebene, ausgerichtet.

Mit den Vektoren  $\overrightarrow{{}^0r_{C_i}^x}$ ,  $\overrightarrow{{}^0r_{C_i}^y}$  und  $\overrightarrow{{}^0r_L^z}$  und der Translation  $\overrightarrow{{}^0p_{W_i}}$  der ursprünglichen  ${}^0T_{W_i}$ , entsteht die Transformationsmatrix

$${}^0T_{C_i} = \begin{pmatrix} \overrightarrow{0r_{C_i}^x} & \overrightarrow{0r_{C_i}^y} & \overrightarrow{0r_L^z} & \overrightarrow{0p_{W_i}} \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} {}^0r_{C_i}^{x_1} & {}^0r_{C_i}^{y_1} & {}^0r_L^{z_1} & {}^0p_{W_i}^x \\ {}^0r_{C_i}^{x_2} & {}^0r_{C_i}^{y_2} & {}^0r_L^{z_2} & {}^0p_{W_i}^y \\ {}^0r_{C_i}^{x_3} & {}^0r_{C_i}^{y_3} & {}^0r_L^{z_3} & {}^0p_{W_i}^z \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (3.4)$$

des Tiefenkartenzuschnitts relativ zum  $K_0$ . Durch Invertierung von  ${}^0T_{C_i}$  und Multiplikation mit  ${}^0T_S$  des 3D-Bildsensors, relativ zu  $K_0$ , wird die gesuchte Transformationsmatrix

$${}^{C_i}T_S = ({}^0T_{C_i})^{-1} \cdot {}^0T_S \quad (3.5)$$

berechnet. Diese Matrix kann sowohl online während des Prozesses als auch zur Datengenerierung in Simulationsszenen eingesetzt werden. Die Ausrichtung des Tiefenkartenzuschnitts kann durch die Definition des Werkstückkoordinatensystems  $K_{W_i}$  beeinflusst werden. Aufgrund der Berücksichtigung des  $K_L$  können auch Zuschnitte in schräg gestellten Kisten mit  $\overrightarrow{0r^z} \nparallel \overrightarrow{0r_L^z}$  erfolgen.

### 3.1.3 Datengenerierung in einer Simulationsszene

Um Daten zu generieren, welche zum Training eines künstlichen neuronalen Netzes zur Verhakungserkennung verwendet werden, wird in dieser Arbeit die Simulationsumgebung CoppeliaSim<sup>1</sup> eingesetzt. Diese Simulationsumgebung zeichnet sich durch vordefinierte Kinematikberechnungen und die Möglichkeit der Implementierung eines bestärkenden Lernverfahrens aus. In der Simulationsszene werden CAD-Daten der Werkstücke und der Kisten maßstabsgetreu nachgebildet und eingefügt. Hierbei besteht die Möglichkeit, durch geringen Aufwand, zwischen unterschiedlichen Ladungsträgern und Werkstückgeometrien zu wechseln.

<sup>1</sup><https://www.coppeliarobotics.com/>

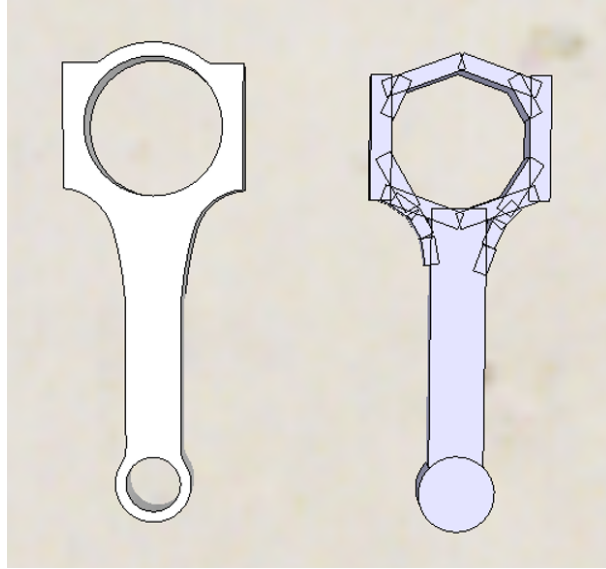


Abbildung 3.4: Zerlegung der konkaven Geometrie einer Pleuelstange in mehrere konvexe Körper

In einem ersten Schritt werden die Werkstückgeometrien aus konvexen Grundkörpern nachgebildet. Die Physik-Engine *Bullet 2.83* wird verwendet, welche die dynamischen physikalischen Ereignisse innerhalb der Simulationsszene simuliert. Diese Physik-Engine hat sich durch empirische Studien als geeignet für folgend beschriebene Prozesse erwiesen. Durch die Zerlegung und Nachbildung der Geometrien als konvexe Volumenkörper werden die Berechnungen beschleunigt. Ein Beispiel einer Zerlegung einer konkaven Pleuelstange ist in Abbildung 3.4 dargestellt. Die Zerlegungen können manuell oder durch einen in CoppeliaSim integrierten *Convex Decomposition* Algorithmus nachgebildet werden.

In der Simulationsszene werden zwei verschiedene Prozesse simuliert: Zuerst wird die Kiste mit den ausgewählten Werkstückgeometrien befüllt, danach erfolgt der Entnahmeprozess und damit die Verhakungserkennung. Abbildung 3.5 stellt den Simulationsablauf zur Datengenerierung für die Verhakungsmethode dar.

### **Befüllung der Kiste:**

Im ersten Prozess wird die Kiste mit bekannten Werkstückgeometrien befüllt. Hierfür werden CAD-Modelle der Kiste und der Werkstücke benötigt. Diese werden in die Simulationsszene geladen. Durch die Größe der Kiste ist auch die Position des Kistenrands definiert und dadurch die möglichen Posen über der Kiste, aus welchen Werkstücke in die

**Funktion: Annotierung für die Verhakungserkennung in der Simulation**

**Eingabe:** CAD-Daten der Kiste und Werkstücke, Anzahl  $n_W$  Werkstücke in der Kiste, Höhenlage  $h_{W_i}$  des Werkstücks in der Kiste, Höhe  $h_K$  der Kiste, Anzahl  $n_K$  zu befüllender Kisten

**Ausgabe:** Annotierung jedes Werkstücks auf Auftreten von Verhakungen

```

0  n = 0
1  solange n < n_K tue
2  |   befülle Kiste mit Anzahl n_W Werkstücken (Ausgangssituation)
3  |   für jedes Werkstück  $W_i$  in der Kiste mit der Höhe  $h_{W_i}$  tue
4  |   |   Werkstück  $W_i$  oberhalb Kistenrand anheben
5  |   |   wenn es existiert ein  $h_{W_j} > h_K$  mit  $j \neq i$  dann
6  |   |   |   rotiere Werkstück um  $x$ -Achse,  $y$ -Achse,  $z$ -Achse
7  |   |   |   wenn es existiert immer noch  $h_{W_j} > h_K$  mit  $j \neq i$  dann
8  |   |   |   |   stelle Ausgangssituation her und annotiere ungültig verhakt
9  |   |   |   |   sonst stelle Ausgangssituation her und annotiere gültig verhakt
10 |   |   sonst stelle Ausgangssituation her und annotiere gültig nicht verhakt
11 |   Kiste für erneute Befüllung leeren
12 |   n = n + 1
13 Ende

```

Abbildung 3.5: Simulationsablauf zur Datengenerierung für die Verhakungserkennungsmethode

Kiste geworfen werden können. Hierzu werden die Werkstückgeometrien über der Kiste in die Szene eingefügt und nacheinander in die Kiste fallen gelassen. Hierbei wird die gesamte Öffnungsfläche der Kiste ausgenutzt und mögliche Abwurfposen der Werkstückmittelpunkte zufällig ausgewählt. Sollte ein Werkstück dennoch nicht in der Kiste landen, wird dies erkannt und ein erneutes Werkstück über der Kiste erstellt und fallen gelassen.

Die Anzahl von möglichen Verhakungen ist abhängig von der Werkstückgeometrie und kann auch beispielsweise beim Transport der Kisten durch Vibrationen entstehen. In der Simulation ist eine Erzeugung von Verhakungen durch Vibrationen allerdings nicht verlässlich genug. Um Verhakungen in der Kiste zu provozieren, kann deshalb definiert werden, ob auch verhakte Werkstücksituationen erstellt und in die Kiste fallen gelassen werden. Hierbei werden zwei Werkstücke gleichzeitig erzeugt und gleichzeitig in die Kiste fallen gelassen. Die Verteilung an verhakten und nicht verhakten Werkstücken kann eingestellt

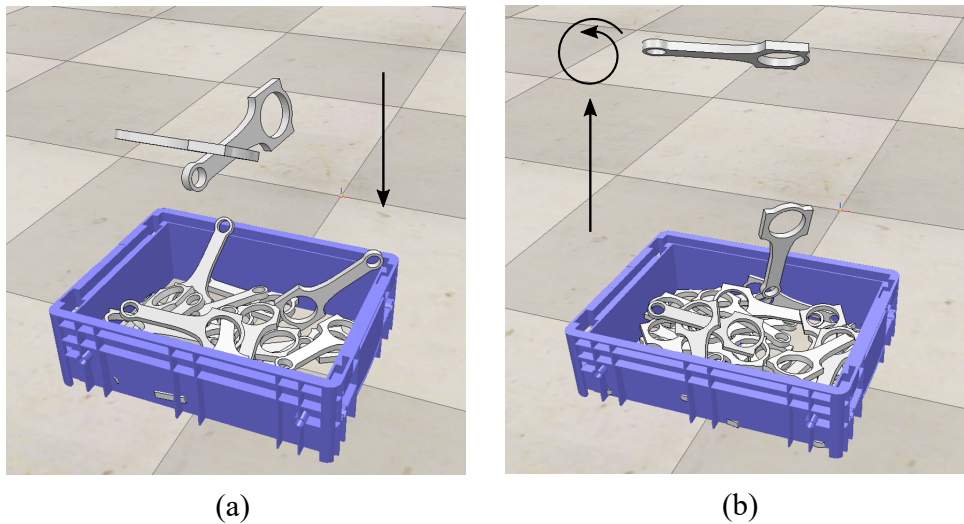


Abbildung 3.6: Simulationsszene in der Software CoppeliaSim: (a) Befüllung der Kiste, (b) Entnahmeprozess des Werkstücks zur Annotierung

werden und wird in Abhängigkeit der Werkstückgeometrie gesetzt. Das Ziel ist es, eine möglichst gleiche Verteilung von verhakten und nicht verhakten Werkstücksituationen zu erzeugen. Mögliche verhakte Ausgangszustände zweier Werkstücke für den Einwurf in die Kiste können in einer CSV-Datei abgespeichert und in der Simulationsszene geladen werden. Die CSV-Datei beinhaltet demnach Werkstückposen zweier Werkstücke, die sichergestellt verhakt sind. Diese CSV-Datei kann im Vorfeld in einer separaten Simulationsszene erstellt werden, die automatisch verhakte Werkstücksituationen aus zwei Werkstücken generiert. Eine mögliche verhakte Situation ist in Abbildung 3.6 (a) dargestellt.

### Entnahmeprozess und Verhakungserkennung:

Nachdem eine Kiste mit einer vordefinierten Anzahl  $n_W$  an Werkstücken befüllt wurde, startet der zweite Prozess. Das Ziel des Prozesses ist es, jedes Werkstück in der Kiste auf eine Verhakung zu prüfen und zu annotieren. Hierfür wird zunächst ein Tiefenkartenzuschnitt eines Werkstückes in der Kistensituation durch einen virtuellen 3D-Bildsensor erstellt. Darauffolgend wird das Werkstück linear über die Kiste angehoben. Nachdem die Endpose über der Kiste erreicht ist, werden die aktuellen Höhenlagen  $h_{W_j}$  im Kistenkoordinatensystem  $K_L$  aller übrigen in der Kistensituation befindlichen Werkstücke, außer der des angehobenen Werkstücks  $h_{W_i}$ , mit einem Grenzwert der Höhenausdehnung der Kiste  $h_G = h_K$  verglichen. Die Reihenfolge der Werkstücke in der Kiste wird durch die

Kistenbefüllung vorgegeben. Das erste Werkstück, welches in die Kiste fallen gelassen wird, wird als erstes angehoben.

Überschreitet ein weiteres Werkstück den oberen Rand der Kiste nach dem Anhebevorgang

$$h_{W_i} > h_G \rightarrow b_{\text{Verhakung}} = \text{true} \quad (3.6)$$

kann eine mögliche Verhakung vorliegen. Durch fehlerhafte Berechnungen durch die Physik-Engine, kann es beim Befüllen der Kiste zu unmöglichen Verhakungssituationen kommen. Um dies zu überprüfen, wird das angehobene Werkstück nacheinander um  $360^\circ$  um alle Koordinatenachsen gedreht, wie in Abbildung 3.6 (b) dargestellt. Durch die Rotationen separiert sich das realistisch verhakte vom angehobenen Werkstück. Das angehobene Werkstück wird mit einer Verhakung annotiert. Die Annotierung erfolgt binär. Ein Pleuelauge kann sich beispielsweise mit einem anderen Pleuelauge, wie in Abbildung 3.7 (a) dargestellt, verhaken. Diese Fehler werden durch die beschriebene Methode erkannt und die Daten nicht im Datensatz zum Training des neuronalen Netzes gespeichert.

Nach erfolgter Annotierung wird die Kistensituation, wie vor dem Anheben, wiederhergestellt. In Abbildung 3.5 wird dies als Ausgangssituation bezeichnet. Danach startet der Prozess erneut und ein anderes Werkstück in der Kistensituation wird angehoben. Dieser Prozess wird für alle Werkstücke in der Kiste durchgeführt, unabhängig davon, ob diese bereits in einer Verhakung involviert waren.

Diese Methode der Annotierung beschränkt die Befüllung der Kiste maximal bis zu deren Rand. Sollten Kisten soweit gefüllt werden, dass Werkstückgeometrien oberhalb des Kistenrandes eine stabile Pose annehmen, so kann der Grenzwert des Höhenvergleichs  $h_G$  manuell größer als die Höhe des Kistenrandes  $h_K$  gesetzt werden.



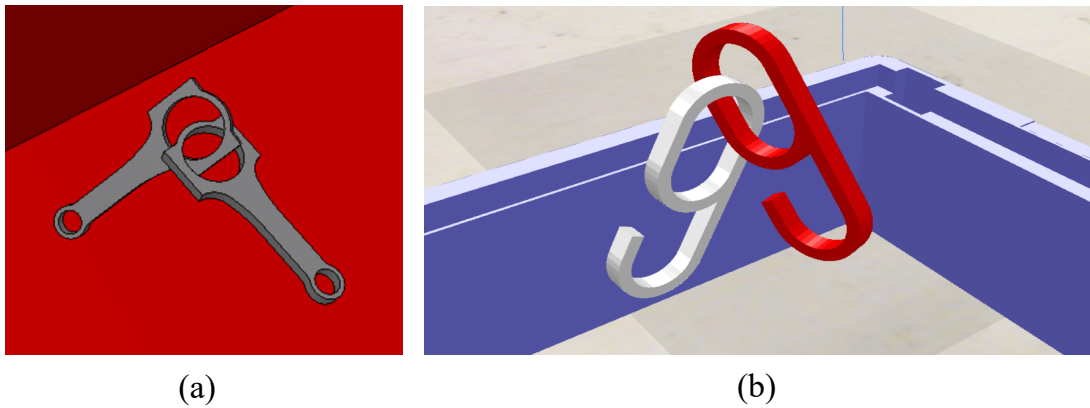


Abbildung 3.7: Unmögliche Verhakungssituation in der Kiste in einer Simulationsszene von (a) Pleuel und (b) Haken. (Moosmann et al. 2020)

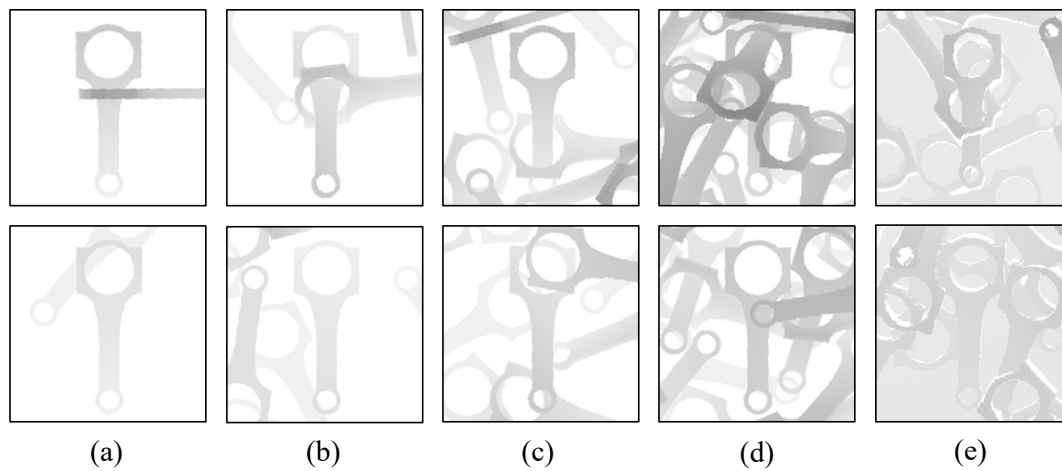


Abbildung 3.8: Tiefenkartenzuschnitte von Pleuelstangen aus der Simulation, jeweils verhakt (oben) und nicht verhakt (unten) mit einer verschiedenen Anzahl von Werkstücken in der Kiste: (a) zwei Werkstücke, (b) fünf Werkstücke, (c) zehn Werkstücke, (d) 20 Werkstücke, (e) reale Tiefenkarten mit zwei bis 20 Werkstücke. (Moosmann et al. 2020)

Abbildung 3.8 zeigt einige Beispiele für annotierte Daten, welche in der Simulationsszene generiert werden. Zu sehen sind jeweils ein verhakter (oben) und ein nicht verhakter Zustand (unten) für eine unterschiedliche Anzahl  $n_W$  an Werkstücken in der Kiste. In Spalte (a) befinden sich nur zwei Werkstücke in der Kiste, während in Spalte (b) bereits fünf Werkstücke vorliegen. Spalte (c) beinhaltet Tiefenkartenzuschnitte aus Kistensituationen mit zehn Werkstücken, Spalte (d) mit 20. In Spalte (e) sind zum Vergleich Tiefenkartenzuschnitte von Kistensituationen dargestellt, welche aus einer realen Roboterzelle stammen.

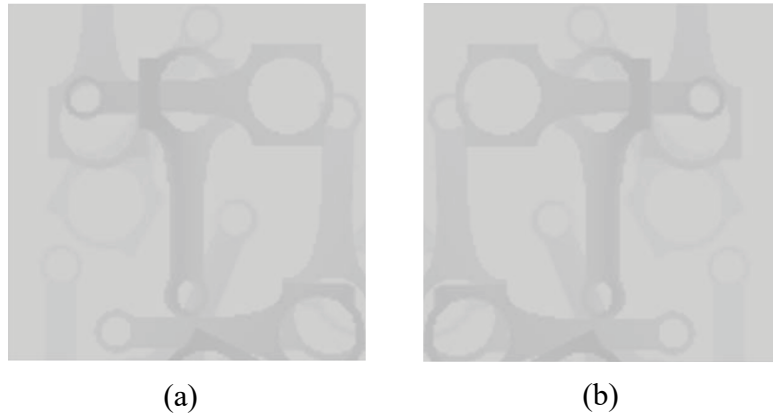


Abbildung 3.9: Horizontale Spiegelung der Tiefenkarte als Augmentierungsmethode: (a) originale Tiefenkarte, (b) horizontal gespiegelte Tiefenkarte

### 3.1.4 Augmentierung der Daten

Unter Augmentierung wird eine Vervielfältigung der Daten verstanden. Im beschriebenen Datensatz sind Datenpaare, bestehend aus einem entlang der  $x$ -Achse des Werkstücks orientierten Tiefenkartenzuschnitts und einer zugehörigen Annotierung, vorhanden. Eine Möglichkeit der Augmentierung dieses Datensatzes besteht in der horizontalen oder vertikalen Spiegelung der Tiefenkartenzuschnitte. Um die gleiche Ausrichtung der Werkstückabbildung in dem Tiefenkartenzuschnitt zu gewährleisten, ist eine horizontale Spiegelung oder eine vertikale Spiegelung mit anschließender Rotation des Tiefenkartenzuschnitts möglich. Ein mögliches Beispiel ist in Abbildung 3.9 dargestellt. Es sind ein originaler Tiefenkartenzuschnitt aus der Simulationsumgebung (a) und die dazugehörige horizontale Spiegelung (b) abgebildet. Die Möglichkeit der Spiegelung ist allerdings nur dann gegeben, wenn eine Spiegelebene in der Werkstückgeometrie definiert werden kann. Sollte die Werkstückgeometrie keine Spiegelebene besitzen, ist eine Augmentierung mittels Spiegelung nicht möglich. Die Werkstücklage kann durch eine Spiegeltransformationsmatrix ebenfalls gespiegelt werden. Da für die Methodik der Verhakungserkennung nur die Bilddaten ausschlaggebend sind, wird hierauf verzichtet.

## 3.2 Transfer von der Simulation in die reale Roboterzelle

In realen Daten der 3D-Bildsensoren treten verschiedene Störfaktoren auf. Trainingsdaten aus Simulationsumgebungen werden hingegen unter idealen Bedingungen erstellt. In der

Bildverarbeitung führen die realen Störfaktoren zu enormen Problemen. Da die neuronalen Netzwerke auf diesen idealen Daten trainiert werden, können sie nur schwer auf reale Störfaktoren generalisieren. Beispiele hierfür sind Abschattungen in der Tiefeninformation, unscharfe oder fehlerhafte Objektlageerkennungen, Rauschen, Helligkeitsunterschiede in der Tiefenbilddarstellung aufgrund anderer Entfernungen zum 3D-Bildsensor oder lückenhafte Bilddaten. Um gleiche Leistungen der neuronalen Netze in der Realität im Vergleich zur Simulationsumgebung herzustellen, werden Transfermethoden eingesetzt, die die idealen Simulationsdaten an die Realität annähern. Im Folgenden werden unterschiedliche Methoden vorgestellt, wobei zwischen Domänenrandomisierung, Inpainting und Domänenanpassung unterschieden wird.

### 3.2.1 Domänenrandomisierung

Bei der Domänenrandomisierung werden Daten so verändert und verrauscht, dass diese den Realdaten ähnlich sind. Die Randomisierungsmethoden basieren hierbei auf analytischen Verfahren, die im Nachgang an eine Datengenerierung erfolgen können. Sie unterscheiden sich von Lernverfahren, wie beispielsweise der Domänenanpassung durch GANs.

#### **Translationsversatz:**

Der Tiefenkartenzuschnitt, eines auf Verhakung zu predizierenden Werkstücks, wird anhand der erkannten Werkstückpose erstellt. Die erkannte Werkstückpose stammt aus der Objektlageerkennung, welche versucht, das CAD-Modell des Werkstücks bestmöglich in die Tiefeninformation einzupassen. Hierbei entstehen Ungenauigkeiten. Um mit diesen Ungenauigkeiten in der realen Applikation umgehen zu können, wird ein Translationsversatz verwendet. Jeder Tiefenkartenzuschnitt wird in der Bildebene verschoben und die entstehenden Ränder ohne Tiefeninformationen schwarz aufgefüllt. Die Translationen sind, in beide Richtungen der Hauptachsen, mit einer Varianz  $\sigma_T$  normalverteilt. Der Mittelwert befindet sich in der Bildmitte.

#### **Rotationsversatz:**

Wie bereits beim Translationsversatz, wird ein Rotationsversatz eingeführt. Hierbei werden die Tiefenkartenzuschnitte normalverteilt mit einer Varianz  $\sigma_R$  um eine Rotationsachse,

senkrecht zur Bildebene zum Bildmittelpunkt, rotiert. Die entstehenden fehlenden Tiefeninformationen werden durch schwarze Pixel in der Tiefenkarte aufgefüllt.

### **Gauß'sches Rauschen:**

In realen Tiefenkarten sind oftmals Kanten und Oberflächen in minderer Qualität dargestellt. Das sogenannte Bildrauschen sind Störungen im Bildsignal. Zudem sind durch die perspektivische Projektion in Tiefenbildern oftmals Abschattungen enthalten. Um diese Störung in optimalen Tiefenkartenzuschnitten aus der Simulationsumgebung zu imitieren, wird ein künstliches Rauschen auf das Bild addiert. Bei kontrastarmen Bildern eignet sich eine Wahrscheinlichkeitsverteilung des Rauschens nach der Gauß'schen Normalverteilung, welche vom Durchschnitt des wahren Bildsignals abhängt. Jeder Pixel erfährt eine Änderung in der Varianz  $\sigma_G$ , wobei die Informationen in dem Tiefenkartenzuschnitt von 0 bis 1 skaliert sind.

### **Helligkeitsrandomisierung:**

Die Helligkeitsrandomisierung wird eingesetzt, um den Umgang mit Werkstücken in unterschiedlichen Höhen in der Kiste robuster zu gestalten. Der Tiefenkartenzuschnitt wird nach der Aufnahme der Tiefeninformation und des Lokalisierens der Kiste auf die Kistenabmessungen normalisiert. Hierbei entspricht der tiefste Wert 0 dem Kistenboden, der höchste Wert 255 dem Kistenrand im 8-Bit Bilddatenformat. Die Helligkeit des Tiefenkartenzuschnitts ist demnach abhängig von der gefundenen Kiste, deren Lage geschätzt und damit mit Toleranzen behaftet ist. Die Randomisierung der Helligkeit wird in einem Wertebereich von 0 bis 255 mit einer Varianz  $\sigma_H$  normalverteilt randomisiert. Der Mittelpunkt ist durch die Helligkeit des jeweiligen Pixels bestimmt.

## **3.2.2 Inpainting**

Lückenhafte Bilddaten entstehen bei einer realen Aufnahme in einer Roboterzelle. Unter der Methode des Inpaintings versteht man ein automatisches Auffüllen lückenhafter Bilddaten. Deshalb wird diese Technik im realen Prozess angewandt und der aufgenommene Tiefenkartenzuschnitt, vor dem Einspeisen in die Verhakungserkennungsmethode, durch Inpainting bearbeitet. Die angewandte Inpaintingtechnologie in Anlehnung an Telea (2004)

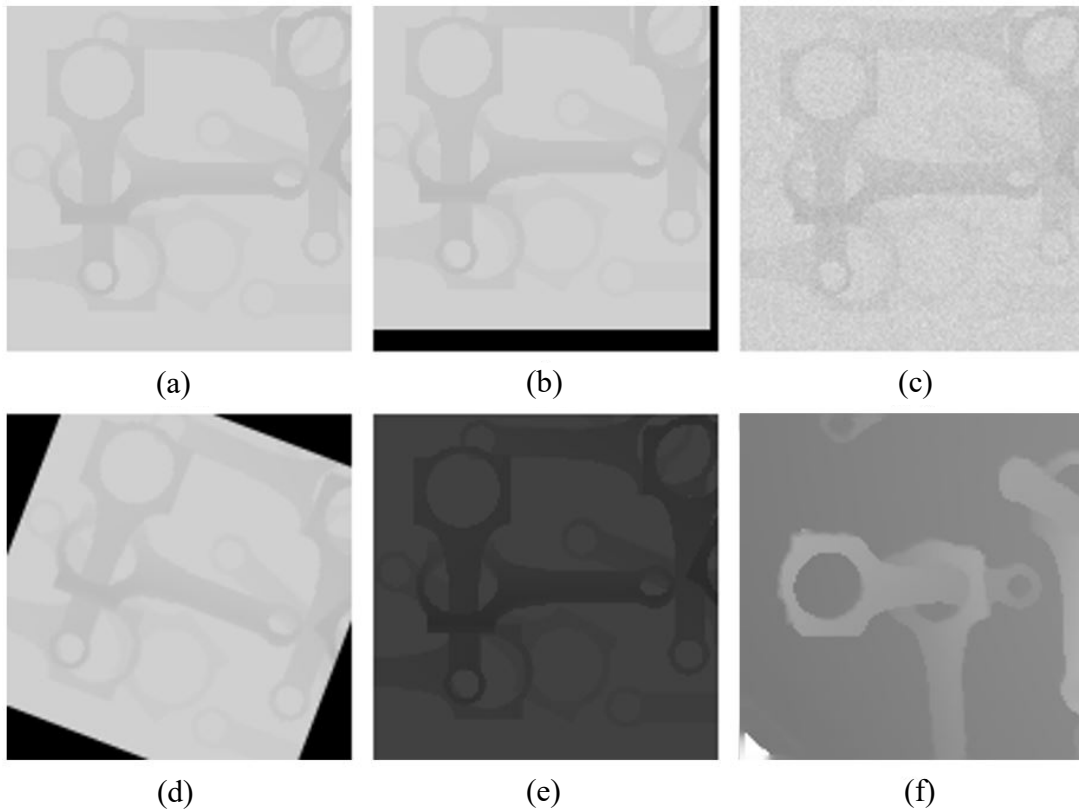


Abbildung 3.10: Transformation einer synthetischen, originalen Tiefenkarte (a) durch unterschiedliche Domänenrandomisierungsmethoden: Translatationaler Versatz (b), Gauß'sches Rauschen (c), Rotatorischer Versatz (d), Helligkeitsveränderung (e), Inpainting einer realen Tiefenkarte (f). (Moosmann et al. 2022b)

füllt die lückenhaften Daten, vom Rand beginnend, und untersucht die direkte Nachbarschaft des fehlenden Pixels, um einen geeigneten Wert für dieses Pixel zu finden.

In Abbildung 3.10 sind alle vorgestellten Domänenrandomisierungsmethoden auf eine ideale Tiefenkarte (a) aus einer Simulationsumgebung angewandt und dargestellt. Translationsversatz (b) sowie Gauß'sches Rauschen (c) sind in der oberen Reihe dargestellt, während der Rotationsversatz (d), die Helligkeitsrandomisierung (e) sowie das Inpainting einer realen Tiefenkarte (f) in der unteren Reihe abgebildet sind.

Die Domänenrandomisierungsmethoden und das Inpainting haben unterschiedliche Einflüsse auf die Performanz einer Methode zur Verhakungserkennung mittels eines neuronalen Netzes. Die Methoden werden sowohl einzeln als auch in Kombinationen verwendet. Die Ergebnisse werden in Kapitel 5 vorgestellt.

### 3.2.3 Domänenanpassung

Unter Domänenanpassung wird das Verändern von Daten aus einer Domäne verstanden, um diese Daten an eine andere Domäne anzunähern. Für die Domänenanpassung wird ein *CycleGAN* nach Zhu et al. (2017) verwendet. Das *CycleGAN* ist ein *Generative Adversarial Network* (GAN), welches für eine Bild-zu-Bild-Übersetzung zwischen zwei Domänen eingesetzt wird. Ein Nachteil eines einfachen GANs, im Vergleich zu einem *CycleGAN*, ist die Notwendigkeit von paarweisen Trainingsdaten. Beispielsweise müsste ein Tiefenkartenzuschnitt eines Werkstücks die gleiche Anordnung und den gleichen Inhalt wie der synthetisch, in Simulation generierte, Tiefenkartenzuschnitt haben. Dies ist im Fall einer Griff-in-die-Kiste-Applikation nicht möglich, da nicht alle Werkstückposen in einer Kisten-situation lokalisiert werden können, um diese synthetisch, in der Simulation, nachzubilden. Ein *CycleGAN* bietet hierfür die Möglichkeit, anhand von wenigen, real aufgenommenen Tiefenkartenzuschnitten, die Unterschiede im Bild zwischen der Simulation und der Realität zu lernen und dieses Wissen auf ungesehene Tiefenkartenzuschnitte anzuwenden. Das implementierte *CycleGAN* für das Verrauschen von Tiefenkartenzuschnitten benötigt rund 1.000 reale Zuschnitte, um zu konvergieren. Die Zeit für die Aufnahme dieser Zuschnitte aus der realen Roboterzelle dauert nur wenige Minuten. Zudem ermöglicht ein *CycleGAN* das Vermeiden eines *Mode Collapse*, welcher sehr oft bei GAN-Trainings auftritt. Dabei lernt das GAN immer die gleichen Bilder mit gleichem Inhalt zu generieren, ohne die Bildinformationen des Eingangsbildes zu berücksichtigen. Das *CycleGAN* schließt dies durch Rücktransformation in das Eingangsbild und Vergleich mit demselben aus.

Abbildung 3.11 zeigt den Aufbau und Ablauf eines *CycleGANs*. Es besteht aus insgesamt vier neuronalen Netzwerken: einem Generator  $G_X$ , der aus der Simulationsdomäne (Domäne X) in die Realitätsdomäne (Domäne Y) transformiert, einem Generator  $G_Y$ , der aus der Realitätsdomäne in die Simulationsdomäne transformiert, einem Diskriminator  $D_X$  und einem Diskriminator  $D_Y$ , welche jeweils versuchen, die gefälschten Tiefenkartenzuschnitte zu identifizieren. Diskriminator  $D_Y$  unterscheidet hierbei zwischen gefälschten Tiefenkarten  $G_X(x)$  und realen Tiefenkarten  $y$  der Domäne Y. Diskriminator  $D_X$  unterscheidet zwischen gefälschten Tiefenkarten  $G_Y(y)$  und simulierten Tiefenkarten  $x$  der Domäne X.

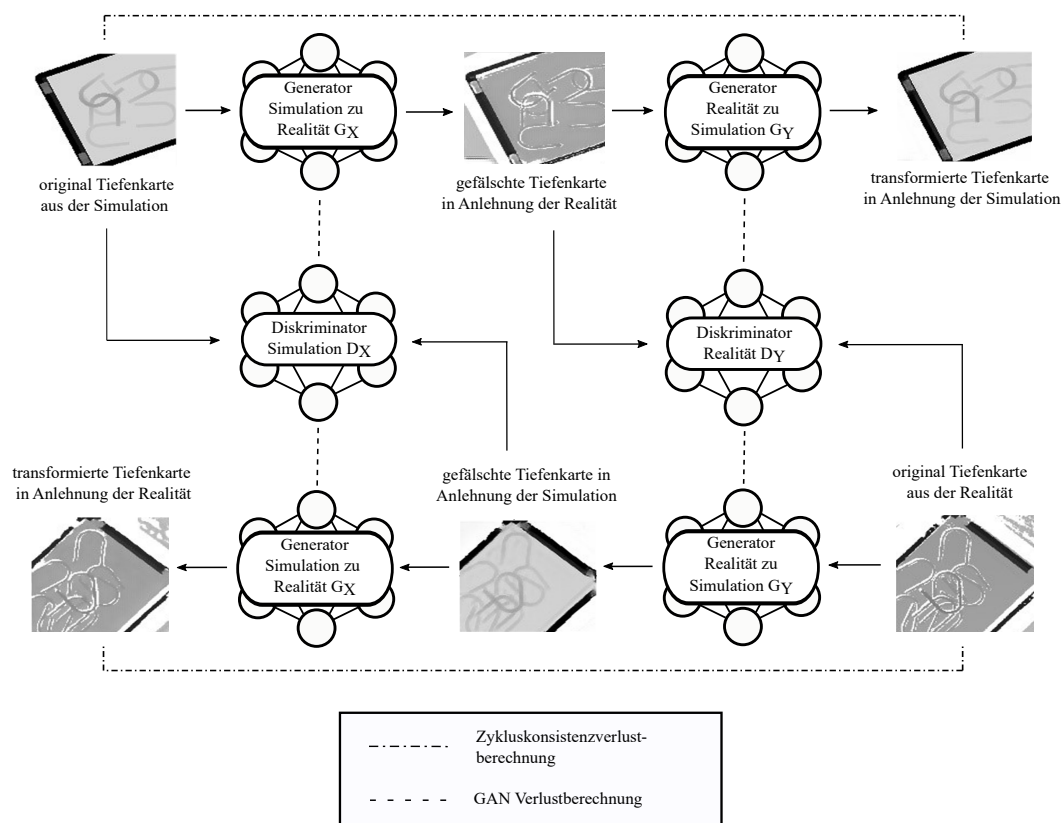


Abbildung 3.11: Aufbau und Ablauf einer Domänenanpassung mittels eines *CycleGANs* in Anlehnung an Moosmann et al. (2022b)

Der Trainingsprozess startet mit der Eingabe eines synthetisch, in der Simulation generierten (links oben) sowie einem realen Tiefenkartenzuschnitt (rechts unten). Generator  $G_X$  versucht nun, den synthetischen Tiefenkartenzuschnitt in einen möglichst realitätsnahen zu transformieren, während Generator  $G_Y$  versucht, von der Realitätsdomäne in die Simulationsdomäne zu übersetzen. Hierbei werden die Generatoren von den Diskriminatoren  $D_X$  und  $D_Y$  überwacht, die lernen, zwischen den Domänen zu unterscheiden und dadurch gefälschte Zuschnitte zu identifizieren. Das Ziel eines Generators ist es, einen Tiefenkartenzuschnitt soweit an die andere Domäne anzupassen, dass der Diskriminator keine Unterschiede wahrnehmen kann.

Die neuronale Netzwerkarchitektur eines Generators ist der von Zhu et al. (2017) vorgeschlagenen nachempfunden, durch den Keras-Tuner optimiert und in Abbildung 3.12 dargestellt. Die Architektur eines Generators beginnt mit der Eingangsschicht, in welche ein  $128 \times 128$  Pixel großer Tiefenkartenzuschnitt eingespeist wird. Nach zwei Downsampling-

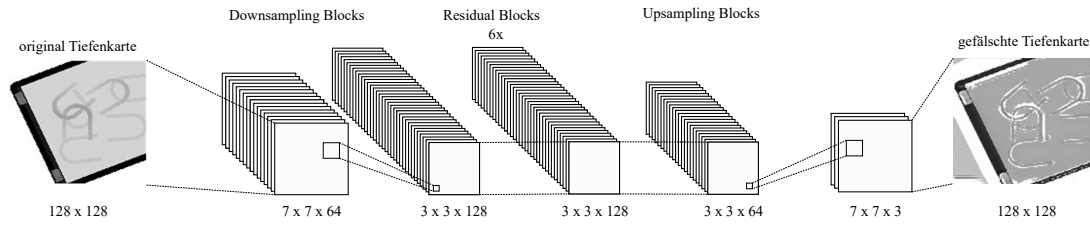


Abbildung 3.12: *CycleGAN* Generator Netzwerkarchitektur (in Anlehnung an Zhu et al. (2017))

schichten mit den Größen  $7 \times 7$  und 64 Merkmalskarten und  $3 \times 3$  mit 128 Merkmalskarten, folgen sechs *Residual Blocks* mit einer Größe von  $3 \times 3$  mit jeweils 128 Merkmalskarten. Die *Residual Blocks* haben den Vorteil, besonders tiefe neuronale Netzwerkarchitekturen realisieren zu können. Diese Eigenschaft ist für das Verändern von Bildern geeignet, um keine relevanten Bilddaten herauszufiltern. Anschließend folgen zwei Upsamplingschichten mit den Größen  $3 \times 3 \times 64$  und  $7 \times 7 \times 3$ , bevor die Ausgangsschicht mit dem gefälschten Tiefenkartenzuschnitt als  $128 \times 128$  Pixel großes Bild folgt. Das Training der Generatoren und Diskriminatoren wird nach folgenden, binären Kreuzentropie-Verlustfunktionen

$$L_{G_X}(G_X, D_Y, X, Y) = \min_{G_X} \max_{D_Y} \log D_Y(y) + \log(1 - D_Y(G_X(x))) \quad (3.7)$$

$$L_{G_Y}(G_Y, D_X, X, Y) = \min_{G_Y} \max_{D_X} \log D_X(x) + \log(1 - D_X(G_Y(y))) \quad (3.8)$$

gesteuert. Die binäre Kreuzentropie vergleicht jede der vorhergesagten Wahrscheinlichkeiten mit der tatsächlichen Klassenausgabe. Sie berechnet dann die Punktzahl, die die Wahrscheinlichkeiten, basierend auf dem Abstand vom erwarteten Wert, bestraft. Ziel der Funktionen ist es, den Verlust des Generators zu minimieren, während der Verlust des Diskriminatoren zu maximieren versucht wird.

Die neuronale Netzwerkarchitektur eines Diskriminatoren (vgl. Abbildung 3.13) startet ebenfalls mit einer Eingangsschicht aus  $128 \times 128$  Pixeln, in welche der jeweilige Tiefenkartenzuschnitt eingespeist wird. Dieser wird durch Faltungsschichten auf  $4 \times 4$  Pixel reduziert, während die Anzahl an Merkmalskarten, zu jeder gefalteten Schicht, verdoppelt wird. Nach



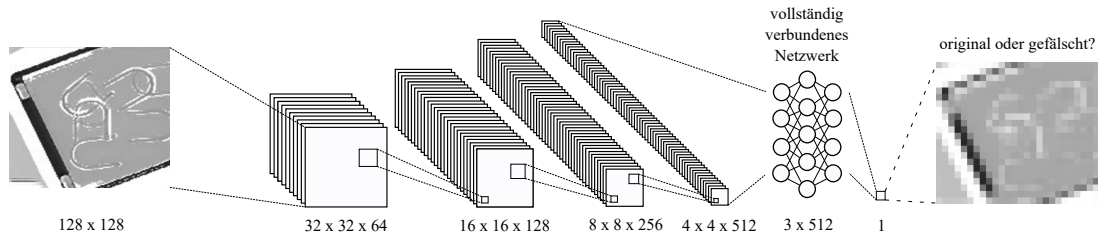


Abbildung 3.13: *CycleGAN* Diskriminator Netzwerkarchitektur (in Anlehnung an Zhu et al. (2017))

insgesamt vier Faltungsschichten folgen drei vollständig verbundene Netzwerkschichten mit jeweils 512 Neuronen. Das Netzwerk wird mit einer Ausgabeschicht mit einer Ausgabe geschlossen, welche angibt, ob der eingegebene Tiefenkartenzuschnitt ein Original oder eine Fälschung ist. Das Training der Generatoren und Diskriminatoren wird mit 1.000 realen und 1.000 synthetischen Tiefenkartenzuschnitten in einer Stapelgröße von einem Datenpaar und mit insgesamt 50 Epochen unternommen. Auf einer NVIDIA Tesla V100 PCIe 32GB GPU benötigt das Training mit einer Lernrate von  $2 \cdot 10^{-4}$  in etwa drei Stunden. Die Lernrate beschreibt, wie stark das Netz die Gewichtung einzelner Neuronen nach jedem Durchlauf anpasst.

Die Besonderheit eines *CycleGANs* ist, dass neben dem Generator- und Diskriminatorverlust, auch ein Zykluskonsistenzverlust berechnet wird. Nachdem die Tiefenkartenzuschnitte in die jeweils andere Domäne transformiert wurden, werden diese wieder in die ursprüngliche Domäne zurücktransformiert. Damit kann sichergestellt werden, dass keine wichtigen Informationen in dem Tiefenkartenzuschnitt durch die Transformationen verloren gegangen sind. Die Berechnung des Zykluskonsistenzverlustes

$$L_{cyc}(G_X, G_Y) = |G_Y(G_X(x)) - x| + |G_X(G_Y(y)) - y| \quad (3.9)$$

erfolgt in einem Pixel für Pixel Bildvergleich, bewertet mit der  $L_1$ -Norm. Ein Trainingszyklus eines *CycleGANs* beinhaltet also eine Transformation von Domäne X in Domäne Y und wieder zurück in Domäne X. Das Training des gesamten *CycleGANs* wird durch eine Gesamtverlustfunktion

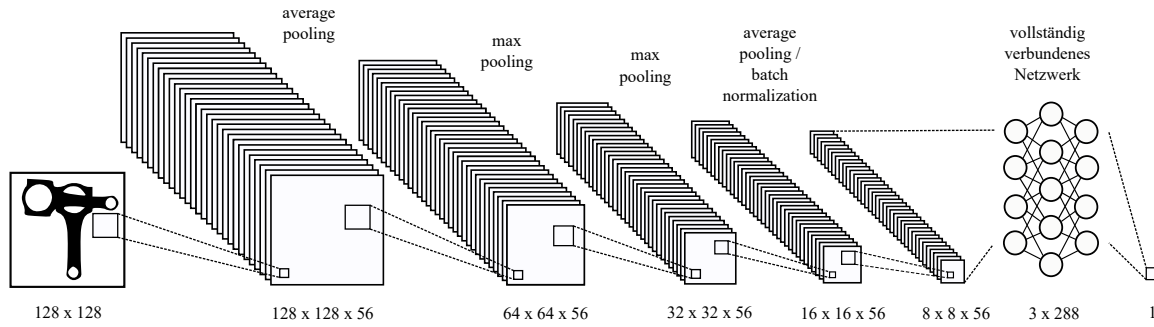


Abbildung 3.14: Netzwerkarchitektur der Verhakungserkennung in Anlehnung an Moosmann et al. (2020)

$$\begin{aligned}
 L_{total}(G_X, G_Y, D_X, D_Y) &= L_{G_X}(G_X, D_Y, X, Y) \\
 &+ L_{G_Y}(G_Y, D_X, X, Y) \\
 &+ L_{cyc}(G_X, G_Y)
 \end{aligned} \tag{3.10}$$

gesteuert, welche sich durch gleichgewichteter Addition aus den Generator- und Diskriminatorverlusten sowie dem Zykluskonsistenzverlust berechnet.

### 3.3 Gefaltete neuronale Netzwerkarchitektur zur Erkennung von Verhakungen

Die gefaltete neuronale Netzwerkarchitektur zur Erkennung von Verhakungen ist in Abbildung 3.14 dargestellt. Sie besteht aus einer Eingabeschicht, mehreren versteckten Schichten und einer Ausgabeschicht und ist einer typischen heruntergetakteten Architektur eines gefalteten neuronalen Netzwerkes nachempfunden. Die Architektur und die Hyperparameter sind durch den Keras-Tuner für die Klasifizierungsaufgabe der Verhakungserkennung optimiert. In die Eingabeschicht wird ein Tiefenkartenzuschnitt in Form eines  $128 \times 128$  Pixel großen Graustufenbildes eingegeben. Es folgen mehrere versteckte Schichten, die Downsampling betreiben. Die erste versteckte Schicht hat eine Größe von  $128 \times 128$  Pixel und besitzt 56 Funktionskarten (engl. *feature maps*). Durch Durchschnittsbündelung (engl. *average pooling*) wird die Größe der Funktionskarten auf  $64 \times 64$  Pixel reduziert. Die folgenden versteckten Schichten bestehen ebenfalls aus 56 Funktionskarten. Durch

Maximalbündelung (engl. *max pooling*) wird die Größe weiter auf  $32 \times 32$  Pixel und  $16 \times 16$  Pixel vermindert. Vor der letzten versteckten Schicht, welche aus 56 Funktionskarten der Größe  $8 \times 8$  Pixel besteht, wird nochmals Durchschnittsbündelung und eine Normalisierung des Stapels (engl. *batch normalization*) betrieben. Vor der Ausgabeschicht sind drei vollständig verbundene Netzwerkschichten (engl. *fully connected layer*) mit jeweils 288 Neuronen angeordnet. Die Ausgabeschicht besteht aus einem einzelnen Neuron, welches die Wahrscheinlichkeit der Verhakung des Eingangesignals ausgibt. Die Aktivierungsfunktion der einzelnen Neuronen ist im gesamten Netzwerk die ReLU-Funktion (engl. *Rectified Linear Unit*).

Das gefaltete neuronale Netzwerk zur Verhakungsprognose wird mit einem Datensatz, bestehend aus einer empirisch ermittelten Anzahl von 17.500 Tiefenkartenzuschnitten von lokalisierten Werkstücken  $W_i$  und dazugehörigen binären Annotierungen  $l_{W_i}$ , in Stapelgrößen (engl. *batch size*) von 256 Datenpaaren, trainiert. Der Validierungsdatsatz beinhaltet 2.500 Datenpaare, während der Testdatensatz 5.000 Datenpaare beinhaltet. Die Daten der Datensätze sind je nach Versuch direkt aus der Simulationsumgebung entnommen, nachträglich domänenrandomisiert oder domänenangepasst. Anhand des Validierungsdatsatzes wird die Abstimmung der Hyperparameter des Modells verwendet, während der Testdatensatz zur Kontrolle des Trainings des Modells am Ende der Modelloptimierungen eingesetzt wird. In einem direkten Transfer, von der Simulation in die reale Roboterzelle, werden nur synthetisch generierte Daten aus einer Simulationsszene in den Datensatz übernommen. Für ein Training mit einem domänenrandomisierten oder domänenangepassten Datensatz werden die jeweils bearbeiteten, synthetischen Daten verwendet. Die Daten im Trainingsdatensatz sind mit einer unterschiedlichen Anzahl an Werkstücken in der Kistensituation randomisiert (vgl. Abbildung 3.8). Verhakte und nicht verhakte Situationen sind im Datensatz gleich verteilt.

Das Training wird in 200 Epochen mit einer Lernrate von  $10^{-3}$  durchgeführt. Eine Epoche benötigt auf einer NVIDIA Tesla V100 PCIe 32GB GPU rund 30 Sekunden. Das Training beläuft sich demnach auf eine Gesamtdauer von rund 100 Minuten. Eine binäre Kreuzentropie-Verlustfunktion

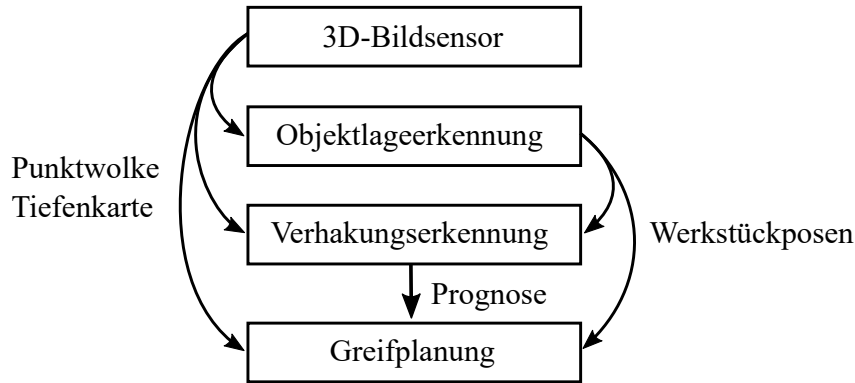


Abbildung 3.15: Ablaufschema und Interaktion einer Verhakungserkennung in einer Griff-in-die-Kiste-Applikation nach Moosmann et al. (2020). In dieser Arbeit wird die Objektlageerkennung nach Palzkill (2014) und die Greifplanung nach Spenrath (2022) verwendet. Die verwendeten Module sind Bestandteil einer Software zur Werkstückvereinzelung, welche in Abbildung 1.1 dargestellt ist.

$$L_{VE} = \sum_{i=1}^{n_W} (l_{W_i} \cdot \log(s_{W_i}) + (1 - l_{W_i}) \cdot \log(1 - s_{W_i})) \quad (3.11)$$

(engl. *binary cross entropy*), oder auch als Log-Verlustfunktion (engl. *log loss*) bezeichnet, steuert das Training mit  $l_{W_i}, s_{W_i} \in [0,1]$ . Ziel ist es, den Verlust zu minimieren. Die Ergebnisse des Trainings und der Verhakungserkennung werden in Kapitel 5 vorgestellt.

### 3.4 Einordnung der Verhakungserkennung in das Gesamtsystem

Das Modul der Verhakungserkennung soll in eine Griff-in-die-Kiste-Software integriert werden und interagiert mit anderen Modulen dieser Software. Abbildung 3.15 beschreibt dies bildlich. Das durch den 3D-Bildsensor aufgezeichnete Tiefenbild, in Form einer Tiefenkarte, wird sowohl der Objektlageerkennung als auch der Verhakungserkennung übergeben. Die Objektlageerkennung nutzt das Tiefenbild zur Erkennung möglicher Werkstückposen. Die Verhakungserkennung nutzt diese Information der Objektlageerkennung und schneidet anhand der gefundenen Werkstückposen Tiefenkartenzuschnitte zu. Diese Tiefenkartenzuschnitte werden einem neuronalen Netzwerk als Eingangswerte übergeben. Das neuronale

Netzwerk prognostiziert für jedes gefundene Werkstück anhand des Tiefenkartenzuschnitts eine Wahrscheinlichkeit, ob das jeweilige Werkstück verhakt ist oder nicht. Die Wahrscheinlichkeit ist ein Konfidenzwert, welcher durch das Neuron in der Ausgangsschicht des neuronalen Netzwerks berechnet wird und entspricht den Kosten  $c_V$ . Hierbei sind höhere Kosten für eine verhakte Situation zu erwarten. Der Kostenfaktor wird mit  $w_V$  gewichtet und mit einem Objektlageerkennungskostenfaktor  $c_O$ , der von einem Algorithmus der Objektlageschätzung stammt und beispielsweise aus Konfidenzwerten der positiven und negativen Kontur, Oberflächen- und Kantenerkennung besteht, zu einem Kostenfaktor des Werkstücks

$$c_{W_i} = c_O \cdot (1 - w_V) + c_V \cdot w_V \quad (3.12)$$

addiert (Spenrath 2022). Der Kostenfaktor des Werkstücks kann in einer heuristischen Greifplanung eingesetzt werden und bewirkt die Priorisierung von Werkstücken, welche ausreichend gut lokalisiert und nicht verhakt sind. Die Struktur der Kostenberechnung ist dabei analog zur Gleichung (2.18) aus Abschnitt 2.3.4.

Der Gewichtungsfaktor  $w_V \in [0,1]$  kann manuell festgelegt werden. Die Werkstückkosten  $c_{W_i}$  werden normalisiert und addiert mit weiteren statischen Kostenfaktoren nach Spenrath (2022) in den Kosten  $c_s$  berücksichtigt. Statische Kostenfaktoren sind beispielsweise die Höhe der Werkstücklage in der Kiste oder die Position relativ zur Kistenmitte. Zusammen mit dynamischen Kostenfaktoren  $c_d$ , welche sich im Laufe der Greifposensuche verändern können, und dem Gewichtungsfaktor  $w_d \in [0,1]$  ergibt sich für die Greifplanung die Heuristikfunktion (2.18) aus Abschnitt 2.3.4.

Die Bewertung der gefundenen Werkstücke anhand der Objektlageerkennungsparameter und der Wahrscheinlichkeit der Verhakung erlaubt der Greifplanung, zuerst nicht verhakte und präzise lokalisierte Werkstücke zu berücksichtigen. Für den Fall, dass in einer Situation nur noch verhakte Werkstücke lokalisiert werden würden, wird in Kapitel 4 dieser Arbeit eine mögliche Enthakungsmethode vorgestellt.

### 3.5 Zusammenfassung

In diesem Kapitel wird eine Methode zur Erkennung von verhakten Werkstücken in einer Kistensituation beim Griff-in-die-Kiste entwickelt. Zu Beginn wird die Ausgangslage und die Aufbereitung benötigter Daten beschrieben, darunter die Daten der Bildsensorik und der Objektlageerkennung. Das Tiefenbild der Kistensituation wird genutzt und anhand der lokalisierten Werkstückposen Tiefenkartenzuschnitte erstellt. Diese bilden die Grundlage für die Verhakungserkennung. Das Training der Lernmethode erfolgt mit synthetisch, in einer Simulationsszene, generierten Daten. In der Simulationsszene werden Kisten befüllt und Werkstücke nacheinander angehoben und auf Verhakungen geprüft. Durch Augmentierung dieser Daten entsteht ein Datensatz, mit dem ein neuronales Netz zur Verhakungserkennung trainiert wird. Die Ausgabe des Netzwerks ist die Prognose der Verhakungswahrscheinlichkeit des Tiefenkartenzuschnitts. Diese Wahrscheinlichkeit wird in einen Kostenfaktor umgerechnet, um in einer heuristischen Greifplanung Werkstücke ohne Verhakung priorisieren zu können. Der Einsatz von Transfermethoden, die synthetische Daten den realen Tiefenbildern annähern, verbessert die Leistungsfähigkeit der Klassifizierungsaufgabe.

# 4 Entwicklung eines Verfahrens zur Enthakung von verhakten Werkstücken

In diesem Kapitel werden drei Verfahren zur Enthakungsplanung von verhakten Werkstücken in Kistensituationen entwickelt. Zu Beginn wird in Abschnitt 4.1 auf die Ausgangslage und Aufbereitung vorhandener Daten eingegangen. Insbesondere werden die Daten der Objektlagererkennung sowie die Daten der Greifplanung beleuchtet. In Abschnitt 4.2 wird eine Repräsentation von Bewegungspfaden zur Enthakung mit einem Hemisphärenmodell dargestellt. Eine Vorstellung von zwei überwachten, maschinellen Lernmethoden zur Auswahl geeigneter Enthakungspfadposen erfolgt in Abschnitt 4.3. Neben den zwei überwachten, maschinellen Lernverfahren wird in Abschnitt 4.4 eine Methode des bestärkenden, maschinellen Lernens beschrieben. Um die Enthakungsmethoden von der Simulationsumgebung in die Realität zu übersetzen, werden die bereits vorgestellten Transfermethoden der Domänenrandomisierung und Domänenanpassung aus Kapitel 3 in Abschnitt 4.5 angewandt. Im Laufe dieser Arbeit werden die drei maschinellen Lernmethoden zur Enthakung miteinander verglichen. Eine Übersicht ist in Abbildung 4.1 dargestellt. In Abschnitt 4.6 wird ein Ablehnungskriterium zur Reduzierung situativ unmöglicher Enthakungsstrategien für alle drei maschinelle Lernmethoden vorgestellt. Die Einbindung des Enthakungsverfahrens in ein Griff-in-die-Kiste-Gesamtsystem wird in Abschnitt 4.7 erläutert. Das Kapitel wird mit einer Zusammenfassung in Abschnitt 4.8 beendet.

## 4.1 Ausgangslage und Aufbereitung vorhandener Daten

In diesem Abschnitt wird die Ausgangslage und Aufbereitung vorhandener Daten erläutert. Neben den bereits bekannten Daten der Bildsensorik und Objektlageschätzung, wie sie bereits in Abschnitt 3.1.1 vorgestellt wurden, kommen für die Enthakungsplanung

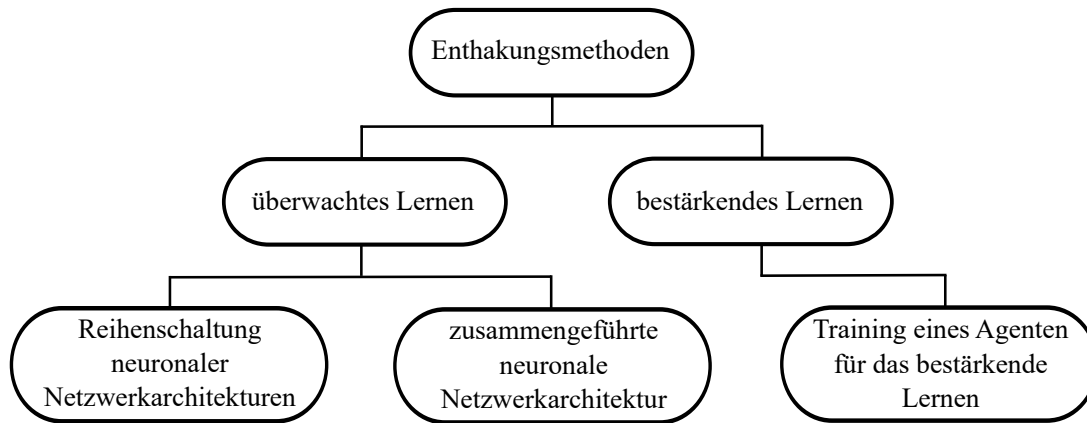


Abbildung 4.1: Übersicht der zu vergleichenden Enthakungsmethoden

Daten der Greifplanung hinzu. Die Enthakungsplanung wird unabhängig von einem konkreten Greifplanungsalgorithmus entwickelt. Die Enthakung wird in dieser Arbeit für die Verwendung mit einem konkreten Greifplanungsverfahren beschrieben, kann aber auch mit anderen eingesetzt werden.

#### 4.1.1 Daten der Bildverarbeitung und Objektlageschätzung

Für die Entwicklung einer Enthakungsmethodik werden Daten der Bildsensorik und der Objektlageschätzung benötigt. Wie bereits für die Verhakungserkennung in Abschnitt 3.1.1 beschrieben, werden die Position und Orientierung des Ladungsträgers  ${}^0T_L$  und der gefundenen Werkstückpose  ${}^0T_W$  relativ zum Weltkoordinatensystem  $K_0$  verwendet. Zu jeder Werkstückpose wird zudem der jeweilige Tiefenkartenzuschnitt benötigt. In Abschnitt 3.1.2 wird die Erstellung dieser Zuschnitte erläutert. Die vollständige Punktwolke, welche durch den 3D-Bildsensor aufgezeichnet wurde, wird ebenfalls im späteren Verlauf für die Kollisionsbetrachtung eingesetzt (vgl. Abbildung 3.1). Die Prognose der Verhakungserkennungsmethode, in Form der Kosten  $c_V$  aus Kapitel 3, wird als Entscheidungsgrundlage verwendet, ob eine Enthakungspfadplanung für eine in einer Kistensituation befindliche Werkstückpose notwendig ist.

#### 4.1.2 Daten der Greifplanung

Zu den bereits bekannten Daten der Bildverarbeitung und Objektlageschätzung werden für die Entwicklung der Enthakungsmethodik zusätzliche Daten der Greifplanung benötigt. Da



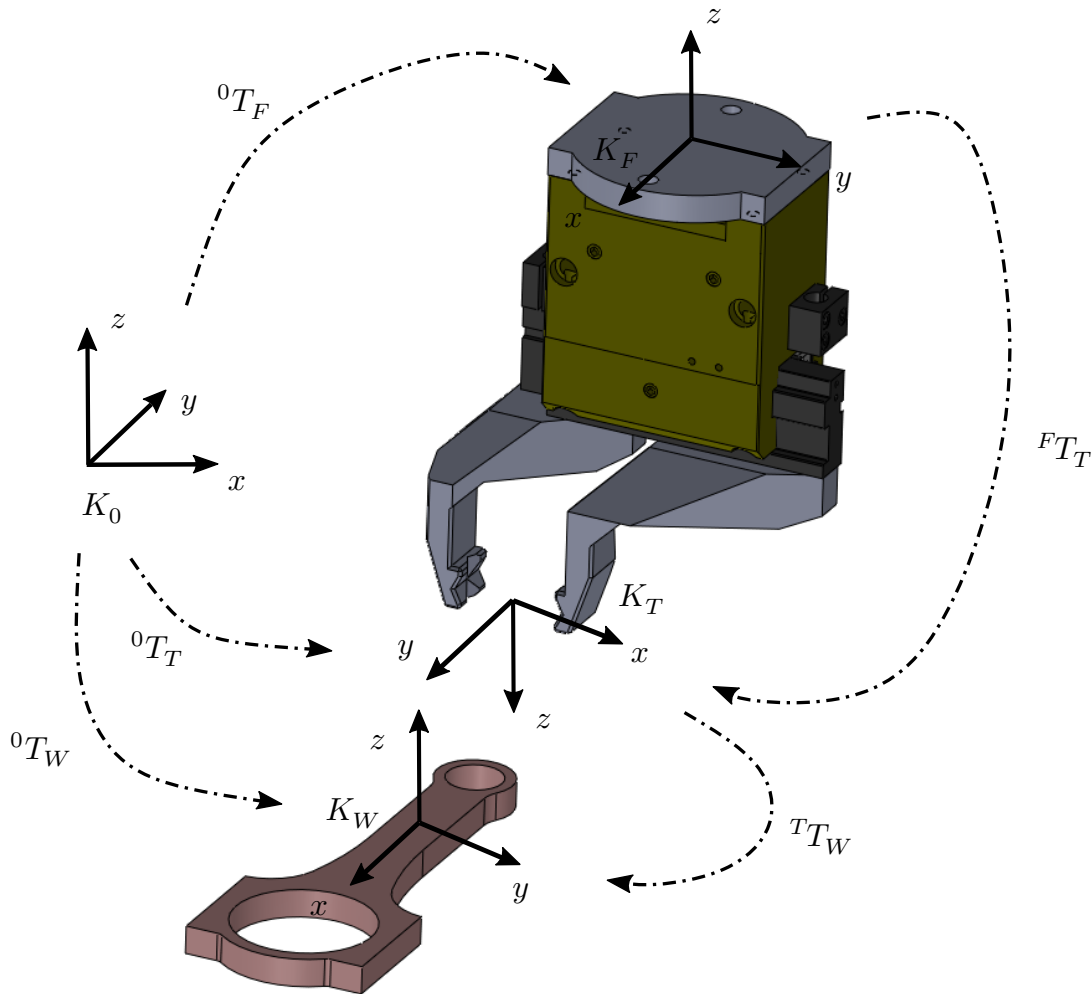


Abbildung 4.2: Übersicht von Koordinatensystemen und Transformationsmatrizen von Roboterflansch, Werkstück, Tool Center Point (TCP) und dem Weltkoordinatensystem

mögliche Enthakungspfade von der Greifergeometrie abhängen können, muss diese in Form eines CAD-Modells zur Verfügung stehen. Abbildung 4.2 zeigt eine solche Greifergeometrie. An diesem Modell werden verschiedene Koordinatensysteme festgelegt, die für die Berechnung möglicher Pfadposen notwendig sind. Hierzu zählt die Montage des Greifers an einem Industrieroboterarm. Die Verbindung zwischen Greiferelement und Roboterarm wird durch einen Flansch realisiert und in Abbildung 4.2 mit dem Koordinatensystem  $K_F$  dargestellt. Zudem wird an der Greifergeometrie ein Werkzeugmittelpunkt definiert, welcher in der englischen Sprache auch als *Tool Center Point*, oder TCP abgekürzt, bezeichnet wird. Das Koordinatensystem des Werkzeugmittelpunkts wird als  $K_T$  definiert.

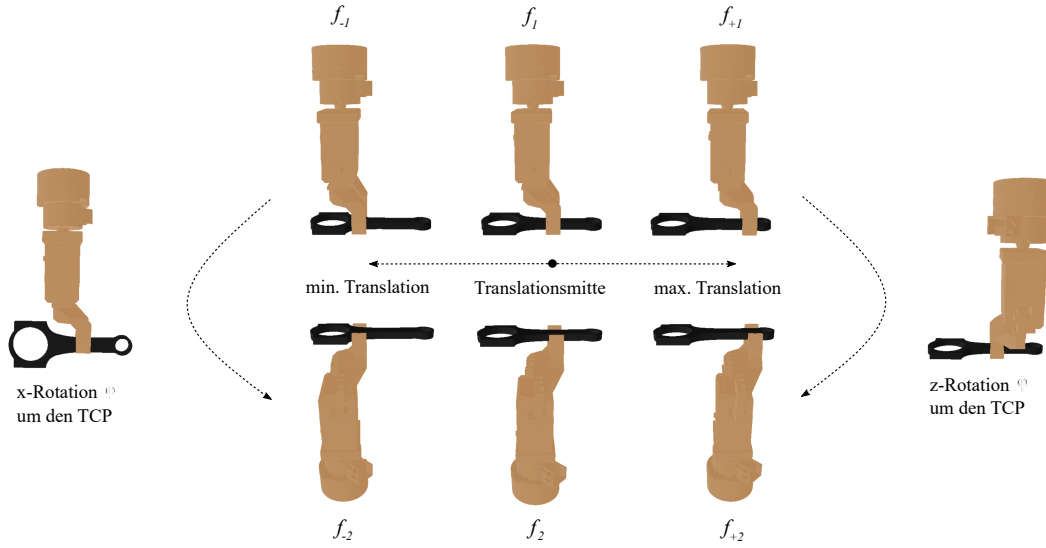


Abbildung 4.3: Mögliche rotatorische und translatorische Freiheitsgrade von definierten Greifposen am Beispiel der Werkstückgeometrie eines Pleuels

Die CAD-Daten des Werkstücks werden bereits für die Objektlageschätzung verwendet. In der Greifplanung können die CAD-Daten zur Definition mehrerer möglicher Greifposen am Werkstück verwendet werden. Die Definition der Greifposen kann manuell oder automatisch sowohl für Sauggreifer als auch für Klemmgreifer (Khalid et al. 2021; Kleeberger et al. 2021) erfolgen. Gespeichert werden die definierten Greifposen in Transformationsmatrizen

$${}^T_f T_W = \begin{pmatrix} \overrightarrow{T_f r_W^x} & \overrightarrow{T_f r_W^y} & \overrightarrow{T_f r_W^z} & \overrightarrow{T_f p_W} \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} T_f r_W^{x_1} & T_f r_W^{y_1} & T_f r_W^{z_1} & T_f p_W^x \\ T_f r_W^{x_2} & T_f r_W^{y_2} & T_f r_W^{z_2} & T_f p_W^y \\ T_f r_W^{x_3} & T_f r_W^{y_3} & T_f r_W^{z_3} & T_f p_W^z \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (4.1)$$

Für die Vervielfältigung der Greifposen am Werkstück entlang der Hauptachsen des  $K_W$  können zudem verschiedene Freiheitsgrade bei der manuellen Greifposendefinition deklariert werden. Mögliche Freiheitsgrade werden mit den Indizes  $f$  und  $\varphi$  deklariert. Diese unterscheiden sich in rotatorische und translatorische Freiheitsgrade. Eine Übersicht über mögliche Freiheitsgrade einer Greifpose wird in Abbildung 4.3 mit einem Pleuel und einer Klemmgreifergeometrie abgebildet. Ein translatorischer Freiheitsgrad wird mit einem Index für die Mittellage  $f_1$ , für eine minimale  $f_{-1}$  und eine maximale  $f_{+1}$  Translation entlang einer Hauptachse des  $K_W$  sowie einer Anzahl an Interpolationsschritten  $n_f$  definiert. Um

die Dauer einer folgenden Simulation zur Enthakungspfadbestimmung zu reduzieren, werden nur die maximalen und minimalen Lagen sowie die Mittellage des Freiheitsgrades für translatorische Freiheitsgrade als mögliche Greifposen betrachtet. Somit berechnen sich aus einem translatorischen Freiheitsgrad drei mögliche Greifposen ( ${}_{f_{-1}}^T T_W$ ,  ${}_{f_1}^T T_W$ ,  ${}_{f_{+1}}^T T_W$ ), die in einer folgenden Simulationsszene randomisiert verwendet werden.

Ein rotatorischer Freiheitsgrad wird mit einem minimalen  $\varphi_-$  und einem maximalen Winkel  $\varphi_+$  sowie einer Anzahl der Interpolationsschritte der Rotation  $n_\varphi$  und einer Rotationsachse definiert. Die Drehwinkelschrittweite

$$\phi = \frac{\varphi_+ - \varphi_-}{n_\varphi} \quad (4.2)$$

für eine Rotationsmatrix  $R_k(\phi)$  wird um die Hauptachse  $k$  des  $K_W$  bestimmt. Somit kann die Transformationsmatrix

$${}_{f_2}^T T_W = R_x(\phi) \cdot {}_{f_1}^T T_W \quad (4.3)$$

von  $f_1$  zu  $f_2$  mit der Rotationsmatrix  $R_k(\phi)$  aus Gleichung (4.2) und  ${}_{f_1}^T T_W$  berechnet werden. In Abbildung 4.3 ist die Greifpose mit dem Index  $f_1$  mit einer Anzahl der Interpolationsschritte  $n_\varphi = 1$ , einem minimalen Winkel  $\varphi_{-1} = 0^\circ$  und einem maximalen Winkel  $\varphi_{+1} = 180^\circ$  um die  $x$ -Achse des  $K_W$  zu einer Greifpose mit dem Index  $f_2$  rotiert.

Rotatorische und translatorische Freiheitsgrade können sich überlagern. Dadurch entstehen die Greifposen mit den Indizes  $f_{-2}$  und  $f_{+2}$ . In Abbildung 4.3 sind zudem zwei weitere rotatorische Freiheitsgrade dargestellt. Bei der linken Darstellung handelt es sich um eine Rotation um die  $x$ -Achse des  $K_W$  um  $\phi = 90^\circ$  mit einer Schrittweite von  $n_\varphi = 1$ . Die rechte Darstellung beschreibt eine Rotation um die  $z$ -Achse des  $K_W$  um  $\phi = 180^\circ$  mit einer Schrittweite von  $n_\varphi = 1$ . Wie bereits bei den translatorischen Freiheitsgraden beschrieben, werden zur Reduzierung der Simulationszeit der Datengenerierung die minimalen, maximalen und die Mittelwerte der rotatorischen Freiheitsgrade berücksichtigt.

## 4.2 Repräsentation von Bewegungspfaden zur Enthakung mit einem Hemisphärenmodell

In diesem Abschnitt wird ein Modell zur Repräsentation von Bewegungspfaden zur Enthakung entwickelt. Das Modell besteht aus Pfadposen zur Enthakung, die auf Hemisphären angeordnet sind. Diese Hemisphären werden über die verhakte Werkstücksituation in der Kiste positioniert. Die Auswahl der Hemisphärenpfadposen wird anhand einer Funktion bestimmt. Die Erzeugung der Hemisphären, sowie deren Anzahl, Art und Anordnung, werden in diesem Abschnitt erläutert und validiert. Am Ende dieses Abschnitts wird das Hemisphärenmodell durch zusätzliche Orientierungsänderungen an den Hemisphärenpfadposen erweitert.

### 4.2.1 Verteilung möglicher Pfadposen auf einer Hemisphäre

Um mögliche Enthakungspfadposen anhand einer verhakten Werkstücksituation zu definieren, wird ein Hemisphärenmodell eingeführt. Dieses Hemisphärenmodell führt zu einer Diskretisierung des Aktionsraums. Kontinuierliche Aktionsräume erschweren das Konvergieren maschineller Lernmethoden zu einem Zielzustand enorm. Zudem wird deutlich mehr Trainingszeit benötigt, um eine solche Konvergenz zu erreichen. Aus diesen Gründen wurde das Hemisphärenmodell zur Diskretisierung des Aktionsraums eingeführt. Ein solches Modell ist bislang, nach dem Wissen des Autors, nicht eingeführt worden und basiert auf empirischen Beobachtungen und Überlegungen. Die erste Hemisphäre wird über die verhakte Werkstücksituation im Koordinatenursprung  $K_W$  des zu greifenden Objekts aufgespannt, entlang der  $x$ -Achse des  $K_W$  orientiert und verläuft dabei parallel zur Bodenebene des Ladungsträgers mit dem Hemisphärenkoordinatensystem  $K_H$ . Abbildung 4.4 stellt dies dar.

Auf einer Hemisphäre sind mögliche Pfadposen verteilt. Der Radius  $r_H$  einer Hemisphäre wird empirisch bestimmt. Die Anzahl  $n_E \in \mathbb{N}$  der Ebenen, auf welchen sich mögliche Pfadposen befinden, ist ebenfalls variabel und wird in einer späteren Auswertung in diesem Kapitel evaluiert. Für jede Ebene, außer der höchsten, werden acht Pfadposen vorgeschlagen. Die Pfadpose der höchsten Ebene befindet sich zentral über dem verhakten Werkstück. Die Anzahl von acht Pfadposen ermöglicht Bewegungen in beiden Richtungen

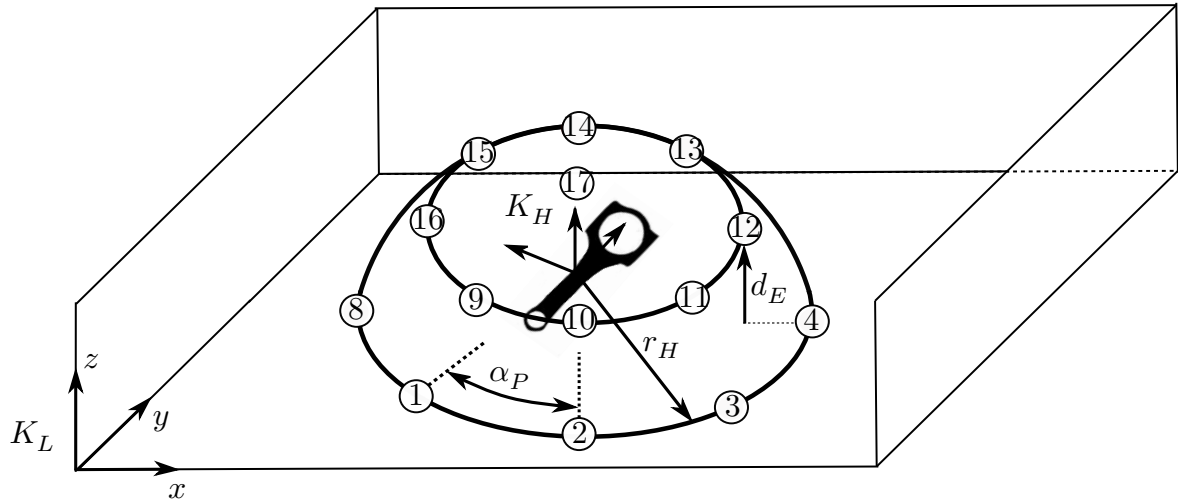


Abbildung 4.4: Übersicht der Verteilung möglicher Pfadposen auf einer Hemisphäre und der Hemisphärenlage relativ zu einem lokalisiertem Werkstück  $W_i$

der Hauptachsen, also der  $x$ -Achse und der  $y$ -Achse des Hemisphärenkoordinatensystems  $K_H$  sowie jeweils um  $45^\circ$  rotierte Zwischenposen. Der Abstand  $d_E$  der Ebenen in einer Hemisphäre wird mittels

$$d_E = \frac{r_H}{n_E} \quad (4.4)$$

bestimmt. Anhand des Abstandes  $d_E$  der einzelnen Ebenen in der Hemisphäre und deren Index  $n_i \in [1, n_E]$  kann die Höhenlage der Ebenen mittels

$$h_E(n_i) = d_E \cdot n_i = \frac{r_H}{n_E} \cdot n_i \quad (4.5)$$

bestimmt werden. Mittels

$$\alpha_P = (n_P - 1) \cdot \frac{\pi}{4}, \forall n_P \in \{1, \dots, 8\} \quad (4.6)$$

wird der Winkel  $\alpha_P$  berechnet, um welchen die Aufteilung der acht Pfadposen auf der Hemisphärenebene im Abstand  $r_H$  um die  $z$ -Achse des Ladungsträgerkoordinatensystems  $K_L$  rotiert wird. Hierfür werden die Indizes  $n_P$  der Pfadposen benötigt. Zur weiteren Berechnung wird eine gleiche Ausrichtung des Ladungsträgerkoordinatensystems  $K_L$  wie das Weltkoordinatensystem  $K_0$  angenommen.

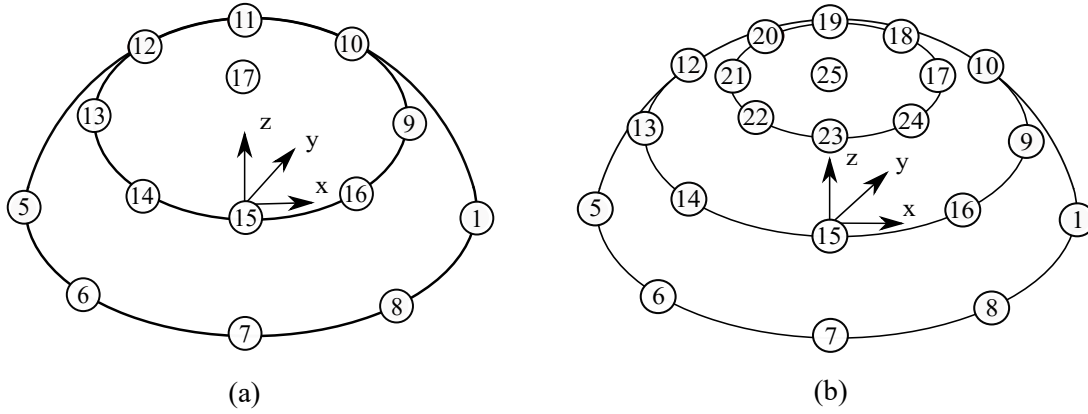


Abbildung 4.5: Verteilung der Hemisphärenpfadposen auf den Hemisphären mit (a) 17 Posen auf 3 Ebenen, (b) mit 25 Posen auf 4 Ebenen nach Moosmann et al. (2021b)

Mit den Ergebnissen aus den Gleichungen (4.4), (4.5) und (4.6) kann nun ein Positionsvektor

$$\vec{p}_{n_i n_P} = \begin{pmatrix} {}^0 p_{W_i}^x \\ {}^0 p_{W_i}^y \\ {}^0 p_{W_i}^z \end{pmatrix} + \begin{pmatrix} \sqrt{r^2 - h_E(n_i)^2} \cdot \cos(\alpha_P) \\ \sqrt{r^2 - h_E(n_i)^2} \cdot \sin(\alpha_P) \\ h_E(n_i) \end{pmatrix} \quad (4.7)$$

zu den auf der Hemisphärenoberfläche verteilten Pfadposen, bestimmt werden. Dieser setzt sich aus einem Positionsvektor des verhakten Werkstücks relativ zum Ladungsträgerkoordinatensystem  $K_L$  addiert mit der Verteilungsberechnung der Pfadposen auf der Hemisphäre zusammen. Die Orientierung des Werkstücks an den Hemisphärenpfadposen wird vorerst wie in der Ausgangssituation beibehalten und erst in einem späteren Schritt in Abschnitt 4.2.5 verändert.

Mit der Kenntnis der Werkstücklage  ${}^0 T_{W_i}$  und den Positionsvektoren der einzelnen Hemisphärenpfadposen ist die Verteilungsberechnung vollständig definiert. Mögliche Verteilungen von 17 oder 25 Hemisphärenpfadposen sind in Abbildung 4.5 dargestellt.

## 4.2.2 Annotierung der Hemisphärenpfadposen

Um eine für eine Enthakung vielversprechende Hemisphärenpfadpose auswählen zu können, bedarf es einer Bewertungsfunktion. Die Bewertung der Pfadposen einer Hemisphäre besteht aus zwei Faktoren, die mit einer unterschiedlichen Gewichtung die Annotierung beschreibt. Hierfür werden nicht nur das zu greifende Werkstück  $W_i$ , sondern auch alle anderen

Werkstücke  $W_j$  in der Kistensituation berücksichtigt. Ferner wird eine Zeitkomponente eingeführt. Diese unterscheidet zwischen der Ausgangssituation  $t$ , in welcher der Greifer das Werkstück zwar gegriffen, aber noch nicht bewegt hat und der Situation  $t + 1$  nach der Bewegung zu einer Hemisphärenpfadpose. Der Positionsvektor des Werkstücks, welches gegriffen, aber noch nicht angehoben oder enthakt wurde und bei welchem der Greifer also noch in der Greifpose steht, wird als  $\overrightarrow{0p_{W_j^t}}$  bezeichnet. Der Positionsvektor der Situation, nachdem das Werkstück durch den Industrieroboterarm und Greifer zu einer möglichen Pfadpose bewegt wurde, wird als  $\overrightarrow{0p_{W_j^{t+1}}}$  deklariert. Daraus ergeben sich die Transformationsmatrizen des Werkstücks vor der Industrieroboterbewegung

$${}^0T_{W_j^t} = \begin{pmatrix} {}^0r_{W_j^t}^{x_1} & {}^0r_{W_j^t}^{y_1} & {}^0r_{W_j^t}^{z_1} & {}^0p_{W_j^t}^x \\ {}^0r_{W_j^t}^{x_2} & {}^0r_{W_j^t}^{y_2} & {}^0r_{W_j^t}^{z_2} & {}^0p_{W_j^t}^y \\ {}^0r_{W_j^t}^{x_3} & {}^0r_{W_j^t}^{y_3} & {}^0r_{W_j^t}^{z_3} & {}^0p_{W_j^t}^z \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (4.8)$$

und nach einer Bewegung zu einer ersten Pfadpose

$${}^0T_{W_j^{t+1}} = \begin{pmatrix} {}^0r_{W_j^{t+1}}^{x_1} & {}^0r_{W_j^{t+1}}^{y_1} & {}^0r_{W_j^{t+1}}^{z_1} & {}^0p_{W_j^{t+1}}^x \\ {}^0r_{W_j^{t+1}}^{x_2} & {}^0r_{W_j^{t+1}}^{y_2} & {}^0r_{W_j^{t+1}}^{z_2} & {}^0p_{W_j^{t+1}}^y \\ {}^0r_{W_j^{t+1}}^{x_3} & {}^0r_{W_j^{t+1}}^{y_3} & {}^0r_{W_j^{t+1}}^{z_3} & {}^0p_{W_j^{t+1}}^z \\ 0 & 0 & 0 & 1 \end{pmatrix}. \quad (4.9)$$

Die Zeitkomponente wird nicht nur für das gegriffene Werkstück, sondern auch für alle anderen Werkstücke in der Kistensituation mit  $\overrightarrow{0p_{W_j^t}}$  und  $\overrightarrow{0p_{W_j^{t+1}}}$ , beziehungsweise  ${}^0T_{W_j^t}$  und  ${}^0T_{W_j^{t+1}}$  eingeführt.

Für den ersten Faktor der Bewertungsfunktion

$$d_{W_j}^p = \sqrt{({}^0p_{W_j^t}^x - {}^0p_{W_j^{t+1}}^x)^2 + ({}^0p_{W_j^t}^y - {}^0p_{W_j^{t+1}}^y)^2 + ({}^0p_{W_j^t}^z - {}^0p_{W_j^{t+1}}^z)^2} \quad (4.10)$$

wird die Verschiebung der Werkstücke  $W_j$  in der Kistensituation betrachtet, welche nicht gegriffen werden. Dieser Faktor wird auch als Faktor der Unruhe bezeichnet. Hierfür

wird die Position dieser Werkstücke vor und nach dem Griff mithilfe einer euklidischen Distanzberechnung verglichen.

Je mehr sich die Werkstücke während der Ausführung der Entnahme und der Enthakungsbewegung bewegen, desto schlechter wird die Pfadpose bewertet. Um hierfür auch die Werkstückrotationen

$${}^{W_j^t}R_{W_j^{t+1}} = \left({}^0R_{W_j^t}\right)^{-1} \cdot {}^0R_{W_j^{t+1}} \quad (4.11)$$

zu berücksichtigen, wird die inverse Rotationsmatrix der Werkstücklage vor der Enthakungsbewegung  ${}^0R_{W_j^t}$  mit der Rotationsmatrix der Werkstücklage nach der Enthakungsbewegung  ${}^0R_{W_j^{t+1}}$  multipliziert.

Die in Gleichung (4.11) entstandene Matrix wird im Folgenden in Drehzeigerdarstellung umgerechnet, um die Rotationen in ein Abstandsmaß zu überführen, welches vergleichbar mit den euklidischen Distanzen ist. Hierfür wird zunächst der Winkel

$$\theta_{W_j} = \cos^{-1} \left( \frac{{}^0r_{W_j^{x_1}} + {}^0r_{W_j^{y_2}} + {}^0r_{W_j^{z_3}} - 1}{2} \right) \quad (4.12)$$

und die Richtung des Drehzeigers in Form eines Vektors

$$\vec{\omega}_{W_j} = \frac{1}{2 \sin \theta_{W_j}} \begin{pmatrix} {}^0r_{W_j^{y_3}} - {}^0r_{W_j^{z_2}} \\ {}^0r_{W_j^{z_1}} - {}^0r_{W_j^{x_3}} \\ {}^0r_{W_j^{x_2}} - {}^0r_{W_j^{y_1}} \end{pmatrix} \quad (4.13)$$

bestimmt. Um die Rotation in Drehzeigerdarstellung mit der Distanz der Verschiebung aus der euklidischen Distanzberechnung zu vergleichen, wird diese im Bogenmaß

$$d_{W_j}^r = \theta_{W_j} \frac{\pi}{180^\circ} \quad (4.14)$$

berechnet.



Aufsummiert ergibt sich somit für das gegriffene und manipulierte Werkstück  $W_i$  aus den Verschiebungen und Rotationen aller anderen Werkstücke  $d_{W_j}^p$  in der Kistensituation die Gesamtdistanz mittels

$$d_{W_i} = \sum_{j=1, j \neq i}^{n_{W_j}} \left( d_{W_j}^p + d_{W_j}^r \right) \quad (4.15)$$

Die Distanzen werden gleichermaßen gewichtet aufsummiert.

In empirischen Versuchen wurde festgestellt, dass die Rotationsbetrachtung aufgrund von Strategien, welche bestärkend gelernt werden, die Funktion gering beeinflusst. Zudem lernt das System bewusst Rotationen durchzuführen, um Werkstücke auszurichten und diese durch eine bekannte Lage durch vorherige Rotation zu enthaken. Diese Rotationen würden durch die Beachtung von  $W_j^t R_{W_j^{t+1}}$  in der Bewertungsfunktion der Hemisphärenpfadposen negativ einfließen. Im Folgenden wird deshalb auf die Betrachtung der Rotationen der nicht gegriffenen Werkstücke verzichtet.

Als zweiter Faktor  $d_{W_{i,v}}$  zur Annotierung der Hemisphärenpfadposen wird die Distanz zwischen dem gegriffenen Werkstück  $W_i$  und den damit verhakten Werkstücken  $W_v$  betrachtet. Es wird zunächst die euklidische Distanz

$$d_{W_{i,v}}^p = \sqrt{({}^0p_{W_i^t}^x - {}^0p_{W_v^t}^x)^2 + ({}^0p_{W_i^t}^y - {}^0p_{W_v^t}^y)^2 + ({}^0p_{W_i^t}^z - {}^0p_{W_v^t}^z)^2} \quad (4.16)$$

zwischen dem gegriffenen Werkstück und einem verhakten Werkstück in der Ausgangssituation vor dem Griff zu Zeitschritt  $t$  berechnet. Danach wird die euklidische Distanz der Werkstücke zum Zeitschritt  $t + 1$

$$d_{W_{i,v}}^{p,t+1} = \sqrt{({}^0p_{W_i^{t+1}}^x - {}^0p_{W_v^{t+1}}^x)^2 + ({}^0p_{W_i^{t+1}}^y - {}^0p_{W_v^{t+1}}^y)^2 + ({}^0p_{W_i^{t+1}}^z - {}^0p_{W_v^{t+1}}^z)^2} \quad (4.17)$$

berechnet, nachdem der Greifer mit gegriffenem Werkstück zu einer ausgewählten Hemisphärenpfadpose bewegt wurde.

Sollten mehrere Werkstücke  $W_v$  mit dem gegriffenen Werkstück  $W_i$  verhakt sein, wird dies als Mehrfachverhakung bezeichnet. Die Mehrfachverhakung kann durch die vorgeschaltete Simulationsszene für die Verhakungserkennung erkannt und die Werkstückgeometrien, welche zu dieser Mehrfachverhakung führen, identifiziert werden (vgl. Abschnitt 3.1.3). Die Distanzen aus Gleichung (4.16) und (4.17) werden mit der Anzahl  $n_{W_v}$  der verhakten Werkstücke mit dem gegriffenen Werkstück zu einem Distanzfaktor

$$d_{W_{i,v}} = \frac{1}{n_{W_v}} \sum_{v=1}^{n_{W_v}} \frac{d_{W_{i,v}}^p}{d_{W_{i,v}}^{t+1}} \quad (4.18)$$

berechnet. Je kleiner  $d_{W_{i,v}}$  ist, desto weiter ist die Trennung der verhakten Werkstücke durch die Bewegung zur ausgewählten Hemisphärenpfadpose erfolgt.

In Abhängigkeit der maschinellen Lernmethode, die verwendet wird, können die Faktoren im Wertebereich  $[0,1]$  normalisiert werden. Dies ist nur in den überwachten maschinellen Lernverfahren möglich, da bei der Datengenerierung alle Hemisphärenpfadposen mit dieser Funktion bewertet und für jede Hemisphäre der Minimalwert  $d_{W_{i,v}}^{min}$  und der Maximalwert  $d_{W_{i,v}}^{max}$  bestimmt werden können. Für das bestärkende Lernen stehen diese Informationen nicht in vollem Umfang zur Verfügung. Eine Normalisierung am Beispiel des Distanzfaktors, auch Enthakungsfortschrittsfaktor genannt, wird mittels

$$d_{W_{i,v}}^n = \frac{d_{W_{i,v}} - d_{W_{i,v}}^{min}}{d_{W_{i,v}}^{max} - d_{W_{i,v}}^{min}} \quad (4.19)$$

berechnet. Beide normalisierten Faktoren werden zu einer Gesamtkostenbewertung der Hemisphärenpfadpose

$$c_{n_P} = d_{W_i}^n \cdot w_P + d_{W_{i,v}}^n \cdot (1 - w_P) \quad (4.20)$$

zusammengefasst. Der Faktor der Unruhe  $d_{W_i}^n$  und der Faktor des Enthakungsfortschritts  $d_{W_{i,v}}^n$  werden mit einem Gewichtungsfaktor  $w_P$  respektive  $(1 - w_P)$  multipliziert. Die Bestimmung des Gewichtungsfaktors  $w_P$  wird in Abschnitt 4.2.4.4 evaluiert.

Je kleiner die Kostenbewertung für eine Hemisphärenpfadpose ausfällt, desto größer ist ihre Eignung für eine erfolgreiche Enthakung der Werkstücksituation.

### 4.2.3 Erzeugung von mehreren Hemisphären zur Enthakungspfadplanung

Das Aufspannen einer Hemisphäre über die verhakte Werkstücksituation in einer Kiste bietet die Möglichkeit, eine zusätzliche Pfadpose in einen Roboterentnahmepfad in einem Abstand  $r_H$  zum Werkstückursprung einzufügen. Nicht für alle Werkstückverha- kungen ist eine zusätzliche Pfadpose ausreichend. Das Hemisphärenmodell bietet hierfür die Möglichkeit, mehrere Hemisphären zu erzeugen und damit weitere mögliche Enthakungspfadposen zu berechnen. In Abbildung 4.6 sind drei mögliche Strategien dargestellt, welche in Abschnitt 4.2.4.3 evaluiert werden. Die Orientierungen aller Hemisphären ist entlang der projizierten  $x$ -Achse des Werkstücks parallel zur Ladungsträgerebene ausgerichtet.

Strategie (a) zur Anordnung weiterer Hemisphären verwendet für jede Hemisphäre den gleichen Radius  $r_H$ . Die zusätzlichen Hemisphären werden an zuvor ausgewählten Pfadposen orientiert. Der Mittelpunkt einer weiteren Hemisphäre ist somit die ausgewählte Pfadpose der vorherigen Hemisphäre.

Strategie (b) nutzt als Mittelpunkt für alle Hemisphären den Werkstückursprung des zu greifenden Werkstücks. Der Radius wird für jede weitere Hemisphäre um den empirischen Faktor 1,5 vergrößert.

Strategie (c) wählt, wie bereits Strategie (a), die am besten bewertete Pfadpose der vorherigen Hemisphäre als Mittelpunkt für eine weitere Hemisphäre aus. Im Gegensatz wird der Radius der folgenden Hemisphären jeweils um die Hälfte des vorherigen Hemisphärenradius reduziert.

Diese drei Strategien werden im folgenden Abschnitt validiert.

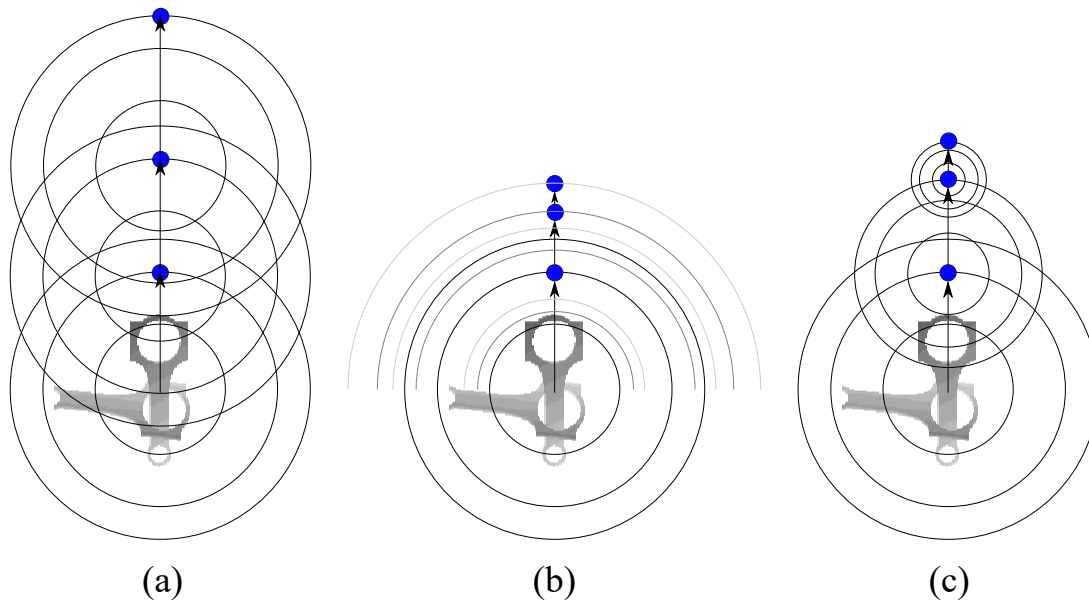


Abbildung 4.6: Enthakungsstrategien (Moosmann et al. 2021b)

#### 4.2.4 Validierung des Hemisphärenmodells von Bewegungspfaden zur Enthakung

In diesem Abschnitt wird das Hemisphärenmodell von Bewegungspfaden zur Enthakung validiert. Zu Beginn wird die optimale Anzahl an Pfadposen in einer Hemisphäre und der Radius der ersten Hemisphäre ermittelt. Ferner wird die Anzahl, die Art und die Anordnung der Hemisphären evaluiert und am Ende des Abschnitts die optimale Gewichtung der Faktoren zur Annotierung der Hemisphärenpfadposen für drei unterschiedliche Werkstückgeometrien, den Rundstahlbügel, das Pleuel und den Haken, ermittelt. Simulationsabbildungen der Werkstücke sind in Abbildung 4.7 dargestellt. Greifposen am Werkstück werden in dieser Validierung nicht betrachtet. Das optimierte Hemisphärenmodell ist die Grundlage für alle folgenden, maschinellen Lernverfahren zur Enthakungspfadplanung.

##### 4.2.4.1 Anzahl der Pfadposen in einer Hemisphäre

Um die optimale Anzahl an Pfadposen in einer Hemisphäre zu bestimmen, wird eine Versuchsreihe durchgeführt. In einer Simulationsszene werden zwei verhakte Werkstücke auf einer horizontalen Ebene abgelegt. Die verhakten Situationen werden im Vorhinein erstellt und nacheinander randomisiert in die Simulationsszene geladen. Um Kollisionen mit Wänden eines Ladungsträgers zu vermeiden, welche die Ergebnisse verfälschen kön-

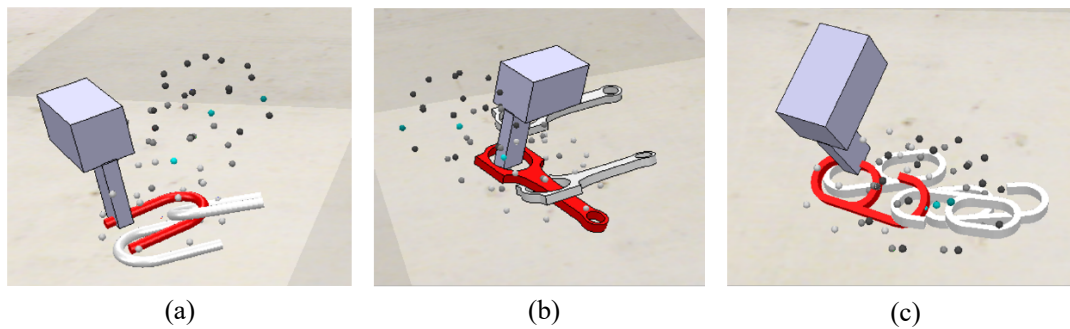


Abbildung 4.7: Simulationsabbildungen der Werkstückgeometrien des (a) Rundstahlbügels, (b) Pleuels und (c) Hakens.

nen, wird bewusst auf einen Ladungsträger verzichtet. Als Werkstückgeometrien werden Rundstahlbügel, Pleuel und Haken untersucht. Nachdem eine verhakte Werkstücksituation erstellt wurde, werden nacheinander alle Pfadposen auf einer Hemisphäre angefahren. Der Radius der Hemisphäre wird mit einem konstanten Wert von 50 Millimetern für alle Werkstückgeometrien deklariert. Die Anzahl an möglichen Hemisphärenpfadposen wird mit 17 und 25 angenommen. Hierfür wird zuerst die Versuchsreihe für 17 Pfadposen, danach für 25 Pfadposen durchgeführt. Zuerst wird nacheinander jede Pfadpose translatorisch angefahren, eine Bewertung durchgeführt und danach die Ausgangssituation wieder hergestellt. Nach jeder Bewegung wird zudem überprüft, ob die Verhakung noch besteht oder gelöst wurde. Die Anzahl an Pfadposen die zu einer Enthakung führen wird aufgezeichnet und als Enthakungsrate relativ zu der Anzahl untersuchter Pfadposen berechnet. Die Anzahl untersuchter Pfadposen ergibt sich aus der Multiplikation der Anzahl an betrachteten, verhakten Werkstücksituationen und der Anzahl von Hemisphärenpfadposen in der ersten Hemisphäre. Die Ergebnisse der Versuchsreihe sind in Tabelle 4.1 dargestellt.

Es zeichnet sich ab, dass bei allen drei Werkstückgeometrien die Anzahl 25 der Hemisphärenpfadposen besser abschneidet. Der Unterschied der Enthakungsrate zu 17 Hemisphärenpfadposen unterscheidet sich allerdings nur minimal um 0,6 bis 1,2 Prozentpunkte. In der Betrachtung der benötigten Zeit, die für die Bewertung der Hemisphärenpfadposen notwendig ist, schneidet die Anzahl von 17 Hemisphärenpfadposen erwartungsgemäß deutlich besser ab. In dieser Versuchsreihe wird nur eine Hemisphäre betrachtet. Sollten weitere Hemisphären in der Auswertung oder Datengenerierung betrachtet werden, steigt die benötigte Zeit für die Datengenerierung exponentiell. In Tabelle 4.2 ist dieses Verhältnis

Tabelle 4.1: Validierung der Anzahl der Hemisphärenpfadposen in einer Hemisphäre anhand der Enthakungsrate bei Bewegungen zu Hemisphärenpfadposen in einer ersten Hemisphäre

Werkstück	Anzahl Hemisphärenpfadposen	Anzahl untersuchter Pfadposen (Anzahl betrachtete verhakte Werkstücksituationen)	Anzahl Pfadposen die zu einer Enthakung führen	Enthakungsrate
Rundstahlbügel	17	16.796 (988)	2.637	15,7 %
	25	24.700 (988)	4.075	16,5 %
Pleuel	17	14.144 (832)	1.429	10,1 %
	25	20.800 (832)	2.226	10,7 %
Haken	17	19.431 (1.143)	972	5,0 %
	25	28.575 (1.143)	1.629	5,7 %

dargestellt. Aufgrund der signifikanten Zeitersparnis, welche durch 17 anstatt 25 Hemisphärenpfadposen entsteht, wird im Laufe dieser Arbeit mit 17 Hemisphärenpfadposen und damit 17 möglichen Bewegungen für jede Hemisphäre fortgefahren.

#### 4.2.4.2 Radius der ersten Hemisphäre

In der vorherigen Versuchsreihe wurde ein konstanter Hemisphärenradius von 50 Millimetern angenommen. In Abhängigkeit der Werkstückgeometrie wird eine Versuchsreihe durchgeführt, um einen optimalen Hemisphärenradius der ersten Hemisphäre zu bestimmen. Hierfür wird eine Simulationsszene genutzt, in welche 1.000 verhakte Werkstücksituationen von zwei verhakten Werkstücken geladen werden. Die verhakten Werkstücksituationen werden vorab bestimmt und randomisiert in der Szene erzeugt. Zudem wird ein Ladungsträger in die Szene geladen, in welchem die Werkstücke verhakt abgelegt werden. Kollisionen mit dem Ladungsträger können identifiziert werden und die Enthakung somit als fehlgeschlagen annotiert werden. Der Ladungsträger (Länge 400 Millimeter, Breite 300 Millimeter, Höhe 148 Millimeter) limitiert somit den Aktions- und Bewegungsbereich und schließt in der Versuchsdurchführung aus, dass eine maximale Bewegung zu einer Hemisphärenpfadpose mit einem maximalen Hemisphärenradius, als optimal deklariert wird. Die Lage der verhakten Werkstücksituation wird über den gesamten Boden des Ladungsträgers randomisiert. Es

Tabelle 4.2: Anzahl an Bewegungen und benötigte Zeit in der Simulation für eine verhakte Werkstücksituation in Abhängigkeit der ausgewählten Aktion

Anzahl Hemi- sphären	Anzahl Hemisphären- pfadposen	Anzahl Bewegungen für eine verhakte Werk- stücksituation	benötigte Simulations- zeit für eine verhakte Werkstücksituation
1	1	1	2,2 s
1	17	17	37,4 s
1	25	25	55,0 s
2	17	289	635,8 s
2	25	625	1.375,0 s
3	17	4.913	10.808,6 s
3	25	15.625	34.375,0 s

wird, wie bereits in der vorherigen Versuchsdurchführung, nur eine Hemisphäre betrachtet, um von den drei möglichen Enthakungsstrategien (a), (b) und (c) unabhängig zu bleiben. Als Ergebnis aus der Optimierung der Anzahl der Hemisphärenpfadposen werden insgesamt 17 mögliche Pfadposen in der Hemisphäre erzeugt. Der Versuch wird mit drei unterschiedlichen Werkstückgeometrien, dem Rundstahlbügel, Pleuel und Haken, durchgeführt. Eins der verhakten Werkstücke im Ladungsträger wird nacheinander zu den 17 Hemisphärenpfadposen in unterschiedlichen Radien bewegt und auf erfolgreiche Enthakung geprüft. Die Radien der Hemisphären werden von 10 Millimeter bis 100 Millimeter untersucht. Es entstehen somit durch einen vorgegebenen Hemisphärenradius mit 17 Pfadposen in einer Hemisphäre und 1.000 verhakten Werkstücksituationen für eine Werkstückgeometrie 17.000 mögliche Pfadposen. Der Anteil an Pfadposen, die zu einer erfolgreichen Enthakung führen, ist das Maß, welches über den optimalen Hemisphärenradius entscheidet. In Abbildung 4.8 ist das Ergebnis der Versuchsreihe dargestellt.

Der maximale Anteil an Pfadposen, die zu einer erfolgreichen Enthakung führen, ist bei der Werkstückgeometrie des Rundstahlbügels bei 50 Millimeter erreicht. Erkennbar ist zudem der starke Abfall der Erfolgsquote, je größer der Radius wird. Dies ist auf einen höheren Anteil an Kollisionen mit den Ladungsträgerwänden zurückzuführen. Bei der Werkstückgeometrie der Pleuel ist ein optimaler Radius bei 60 Millimeter erreicht. Der bestmögliche Radius für Haken liegt bei 20 Millimeter. Die Ergebnisse zeigen, dass für

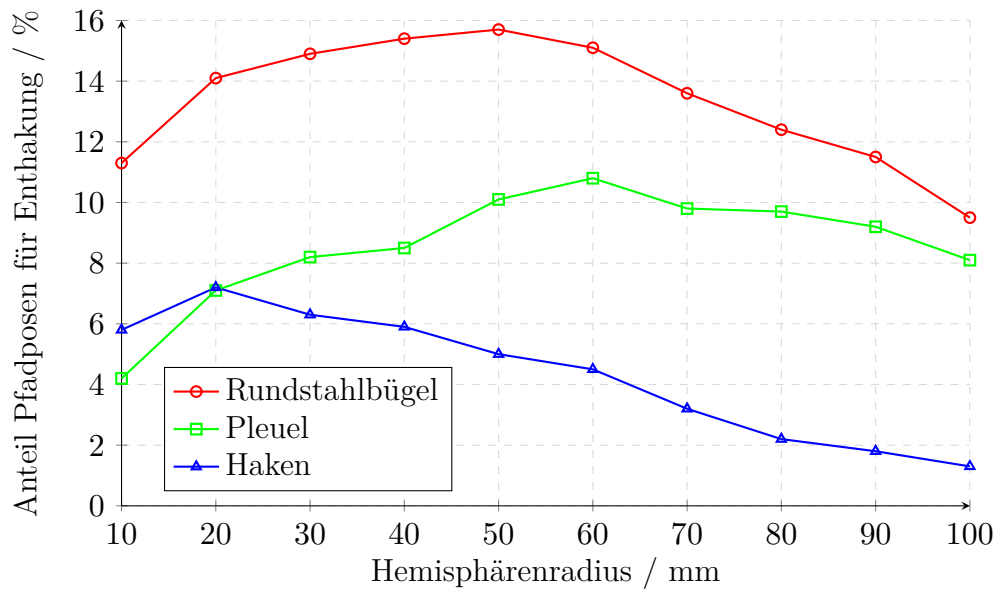


Abbildung 4.8: Bestimmung des Hemisphärenradius anhand des Anteils von Pfadposen für eine erfolgreiche Enthakung für die Werkstückgeometrien eines Rundstahlbügels, Pleuels und Hakens.

neue Werkstückgeometrien der optimale Radius einer Hemisphäre bestimmt werden muss und keine Rückschlüsse aus bisherigen Ergebnissen genutzt werden können. Die Größe der Werkstückgeometrien des Pleuels und des Hakens sind in den beschriebenen Versuchen nahezu gleich, der dazugehörige optimale Hemisphärenradius unterscheidet sich allerdings um 40 Millimeter. Erklärbar ist dieses Verhältnis mit der Geometrie des Werkstücks. Bei den Haken wird meist eine Verhakung mit der offenen Bogengeometrie mit dem Bogenradius von 25 Millimetern beobachtet, während bei den Pleuel, durch die geschlossene Geometrie des Pleuelauges, ein längerer Weg und damit größerer Hemisphärenradius zurückgelegt werden muss, bevor die Situation enthakt ist. Die Ergebnisse dieses Versuches werden im Laufe dieser Arbeit verwendet.

#### 4.2.4.3 Anzahl, Art und Anordnung der Hemisphären

In Abschnitt 4.2.3 wurden drei Strategien vorgestellt, die beschreiben, wie mehrere Hemisphären angeordnet werden können. Diese Strategien werden in folgendem Versuch evaluiert. In einer Simulationsszene werden nacheinander 1.000 verhakte Werkstücksituationen aus zwei Werkstücken ohne einen Ladungsträger erstellt. Für eine verhakte Werkstücksituation werden alle drei möglichen Enthakungsstrategien durchgeführt. Somit bleibt die Situation



Tabelle 4.3: Vergleich der Strategien zur Enthakung für drei unterschiedliche Werkstückgeometrien

Werkstück	Strategie	Enthakungsrate bei Anzahl an Hemisphären / %					Durchschnitt Pfadlänge / mm
		1	2	3	4	5	
Rundstahl- bügel	(a)	15,7	63,7	<b>100,0</b>	100,0	100,0	<b>68,7</b>
	(b)	15,7	35,8	93,6	100,0	100,0	83,5
	(c)	15,7	43,1	98,2	100,0	100,0	92,1
Pleuel	(a)	10,1	56,6	<b>100,0</b>	100,0	100,0	<b>100,0</b>
	(b)	10,1	29,8	56,0	85,5	100,0	112,5
	(c)	10,1	41,9	67,3	94,1	100,0	145,2
Haken	(a)	5,0	62,2	98,1	<b>100,0</b>	100,0	63,3
	(b)	5,0	9,7	15,1	18,8	20,0	91,7
	(c)	5,0	7,1	9,5	13,8	16,7	<b>58,0</b>

für alle drei Strategien vergleichbar. Es wird jede Hemisphärenpfadpose angefahren und nach der Bewegung auf Enthakung geprüft. Von jeder Hemisphärenpfadpose der ersten Hemisphäre wird eine zweite Hemisphäre mit 17 Hemisphärenpfadposen erstellt, außer bei Strategie (b), in welcher die Ursprungspose der Hemisphären gleich bleibt. Dieser Prozess wiederholt sich bis zu maximal fünf Hemisphären. Sollte die verhakte Situation nach fünf Hemisphären noch immer nicht aufgelöst sein, so wird der Prozess für diese Werkstücksituation abgebrochen. Die Enthakungsrate wird in Abhängigkeit der Hemisphärenanzahl für jede Werkstückgeometrie und Strategie ermittelt. Zudem wird die durchschnittliche Enthakungspfadlänge berechnet. Der Versuch wird mit drei unterschiedlichen Werkstückgeometrien durchgeführt. Die Ergebnisse dieses Versuchs sind in Tabelle 4.3 dargestellt.

Für alle drei Werkstückgeometrien schneidet die Strategie (a) am besten ab. Bei Rundstahlbügeln wird eine Enthakungsrate von 100,0% bereits nach drei Hemisphären durch Strategie (a) erreicht. Gleiches gilt für Pleuel. Zudem beträgt die durchschnittliche Pfadlänge bei beiden Werkstückgeometrien mit Strategie (a) den kleinsten Wert mit 68,7 Millimetern bei Rundstahlbügeln und 100,0 Millimetern bei Pleuel. Bei den Haken wird eine Enthakungsrate von 100,0% nach vier Hemisphären mit Strategie (a) erreicht. Bereits bei drei Hemisphären beträgt diese 98,1%. Strategie (b) und (c) erreichen auch nach fünf

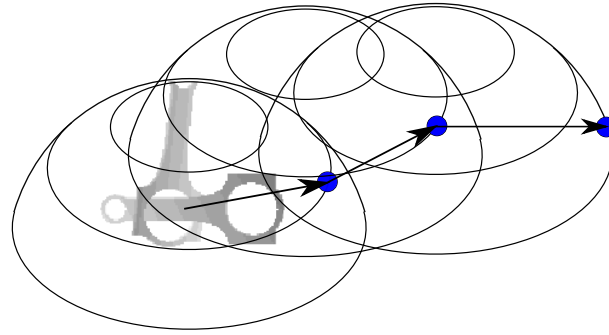


Abbildung 4.9: Dreidimensionale Ansicht einer Enthakungstrajektorie mit drei Hemisphären (Moosmann et al. 2021b)

angefahrenen Hemisphären keine vollkommene Enthakungsrate. Strategie (c) erreicht bei den aufgezeichneten, erfolgreichen Enthakungen eine durchschnittliche Enthakungspfadlänge von 58,0 Millimetern, welche sich allerdings nur gering von Strategie (a) mit 63,3 Millimetern unterscheidet.

Eine Enthakungsrate von 100,0% wird durch Strategie (a), im Gegensatz zu Strategie (b) und (c), für alle Werkstückgeometrien erreicht. Aufgrund dieser Ergebnisse schneidet Strategie (a) für alle getesteten Werkstückgeometrien am besten ab und wird für die weiteren Versuche in dieser Arbeit ausgewählt. Die maximale Anzahl an Hemisphären, die benötigt werden, um eine Enthakung erfolgreich durchzuführen, wird für alle drei Werkstückgeometrien mit drei ermittelt. Obwohl eine Enthakungsrate von 100,0% bei den Haken erst nach vier Hemisphären erreicht wird, wird eine Enthakungsrate von 98,1% nach drei Hemisphären, im Hinblick auf die benötigte Dauer der Datengenerierung und Enthakungspfadlängenerhöhung, als ausreichend angenommen. Abbildung 4.9 stellt eine dreidimensionale Ansicht der ausgewählten Strategie (a) mit drei Hemisphären dar.

#### 4.2.4.4 Gewichtung der Faktoren zur Auswahl der Hemisphärenpfadposen

In Abschnitt 4.2.2 wurden zur Auswahl der Hemisphärenpfadposen Bewertungsfunktionen vorgestellt. Der Faktor der Unruhe im Ladungsträger  $d_{W_i}^n$  und der Faktor des Enthakungsfortschritts  $d_{W_i,v}^n$  werden mit einem Gewichtungsfaktor  $w_P$  respektive  $(1 - w_P)$  multipliziert. Dieser Gewichtungsfaktor wird in einer Versuchsreihe bestimmt. Hierfür wer-

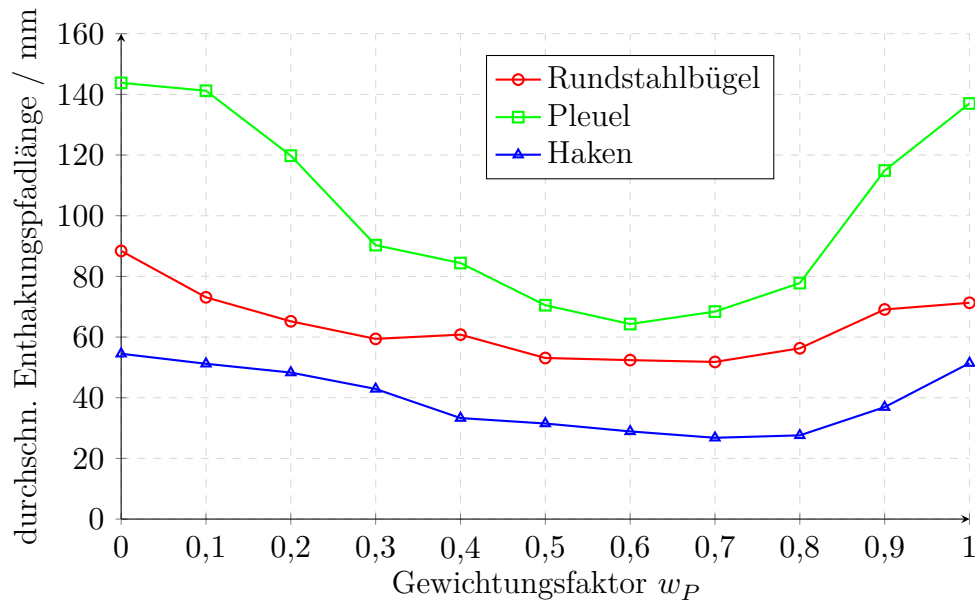


Abbildung 4.10: Bestimmung des Gewichtungsfaktors  $w_P$  der Bewertungsfunktion zur Auswahl einer Hemisphärenpfadpose anhand der durchschnittlichen Enthakungspfadlänge für die Werkstückgeometrien eines Rundstahlbügels, Pleuels und Hakens.

den 100 randomisierte Werkstücksituationen bestehend aus zwei verhakten Werkstücken in eine Simulationsszene ohne Ladungsträger geladen. Untersuchte Werkstückgeometrien sind hierbei die Rundstahlbügel, Pleuel und Haken. Eines der beiden Werkstücke in der Szene wird nacheinander zu 17 Hemisphärenpfadposen mit dem Hemisphärenradius aus Abschnitt 4.2.4.2 bewegt. Dabei wird jede Hemisphärenpfadpose mit der beschriebenen, zusammengeführten Bewertungsfunktion mit einem bestimmten  $w_P$  bewertet. Nach der Bewertung wird die am besten bewertete Hemisphärenpfadpose aus der ersten Hemisphäre zum Ursprungspunkt von der zweiten Hemisphäre, in welcher wieder alle 17 Hemisphärenpfadposen bewertet werden. Dieser Prozess wird so lange durchgeführt, bis drei Hemisphären erreicht wurden. Nach jeder Bewegung wird überprüft, ob eine Enthakung stattgefunden hat. Nachdem eine Enthakung identifiziert wurde, wird die verhakete Werkstücksituation wiederhergestellt und der Prozess startet mit einem anderen Gewichtungsfaktor erneut. Da der Radius in der verwendeten Strategie (a) zur Enthakung konstant bleibt, kann der kürzeste Enthakungspfad anhand der Enthakungspfadlänge identifiziert werden. Somit wird für eine verhakete Situation der Gewichtungsfaktor  $w_P$  für den effizientesten Enthakungspfad bestimmt. Für alle 100 randomisierten Werkstücksituationen wird der Durchschnitt der effizientesten Enthakungspfadlängen ermittelt. Die entstehenden Werte in Abhängigkeit

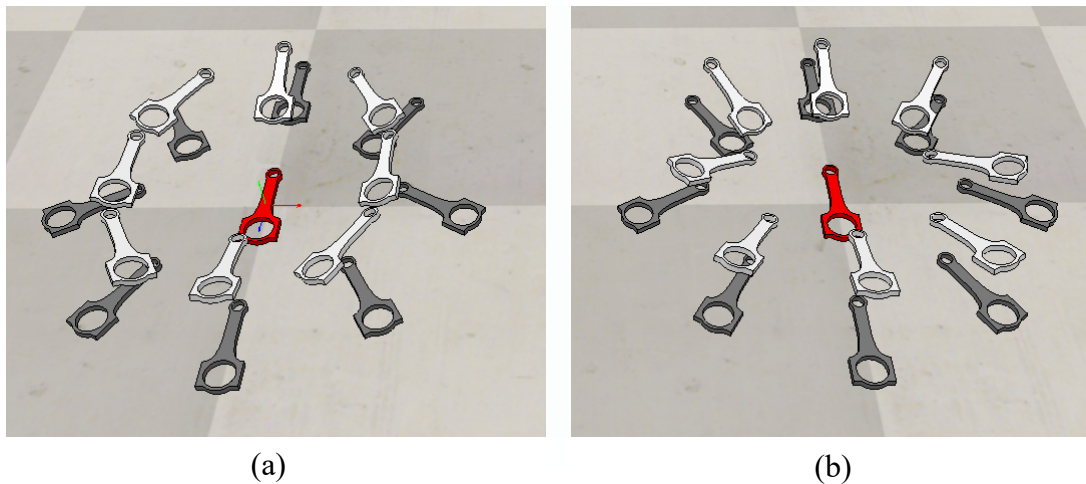


Abbildung 4.11: Rotationsmodell (a) und (b)

der unterschiedlichen Gewichtungsfaktoren und Werkstückgeometrien sind in Abbildung 4.10 abgebildet.

Für Rundstahlbügel und Haken ergibt sich ein optimaler Gewichtungsfaktor  $w_P$  von 0,7 bei einer durchschnittlichen Enthakungspfadlänge von 51,8 Millimetern respektive 26,8 Millimetern. Bei Pleuel liegt der optimale Gewichtungsfaktor bei 0,6 mit einer durchschnittlichen Enthakungspfadlänge von 64,3 Millimetern. Für neue Werkstückgeometrien muss der Gewichtungsfaktor neu ermittelt werden. Es ist davon auszugehen, dass er sich auf 0,6 oder 0,7 beläuft. Diese Werte für  $w_P$  werden im weiteren Verlauf der Arbeit verwendet.

#### 4.2.5 Orientierungsänderung an den Hemisphärenpfadposen

Das Hemisphärenmodell kann durch Orientierungsänderungen an den Hemisphärenpfadposen erweitert werden. Diese können zusätzlich aktiv geschaltet werden. Die Orientierungsänderungen folgen zwei Rotationsmodellen. Diese sind in Abbildung 4.11 abgebildet. Um den Aktionsraum des Hemisphärenmodells nicht zu vergrößern, wird jeder Hemisphärenpfadpose nur eine Werkstückorientierung zugewiesen. Diese Werkstückorientierung wird durch Rotationen des Werkstücks entlang der Koordinatenachsen des Hemisphärenkoordinatensystems  $K_H$  erzeugt. Hierbei wird zuerst um die  $x$ -Achse, danach um die  $y$ -Achse und zuletzt um die  $z$ -Achse mit den Rotationsmatrizen  $R_x(\alpha(n_P))$ ,  $R_y(\beta(n_P))$  und  $R_z(\gamma(n_P))$  rotiert. Die Berechnung der Transformationsmatrix der Werkstücklage  ${}^0T_W$  im Weltkoordinatensystem

$K_0$  zur neuen rotierten Lage an einer bestimmten Hemisphärenpfadpose berechnet sich aus dem Positionsvektor  $\overrightarrow{^0p_{n_i n_P}}$  der Pfadpose und der kombinierten Rotationsmatrix  $R_{n_P}$ . Die kombinierte Rotationsmatrix wird mittels

$$R_{n_P} = R_z(\gamma(n_P)) \cdot R_y(\beta(n_P)) \cdot R_x(\alpha(n_P)) \quad (4.21)$$

berechnet. Die Werte der Winkel  $\alpha(n_P)$ ,  $\beta(n_P)$  und  $\gamma(n_P)$  in Abhängigkeit zur Nummer der Hemisphärenpfadposen  $n_P$  sind in Anhang A.1 gelistet.

Die Rotationsmodelle sind empirisch aus Beobachtungen realer Entnahmeversuche entstanden. Beide Rotationsmodelle unterscheiden sich nur in der zweiten Ebene der Hemisphärenpfadposen. In der untersten Hemisphärenebene, also für die Hemisphärenpfadposen 1 bis 8, sind die Orientierungsänderungen in beiden Rotationsmodellen gleich. Das Werkstück lässt sich in dieser Ebene nur um die  $z$ -Achse des  $K_H$  rotieren, da sonst eine Kollision mit dem Ladungsträgerboden riskiert wird. Hierbei wird das Werkstück in die Bewegungsrichtung ausgerichtet, sodass beispielsweise das Pleuelauge in Richtung der anzufahrenden Hemisphärenpfadpose zeigt. Die zweite Ebene beinhaltet zusätzlich Rotationen um weitere Koordinatenachsen. Während in Rotationsmodell (b) durch eine zusätzliche Rotation um die  $y$ -Achse des Werkstücks versucht wird, das Werkstück weiterhin in Bewegungsrichtung der Hemisphärenpfadpose auszurichten, wird in Rotationsmodell (a) versucht, das Werkstück quer zur Bewegungsrichtung auszurichten. Hemisphärenpfadpose mit der Nummer 17 ist in beiden Rotationsmodellen gleich und befindet sich im Abstand  $r_H$  direkt über dem Werkstück im Ursprungszustand, was bedeutet, dass nur eine Translation entlang der  $z$ -Achse des  $K_H$  ausgeführt wird. In einer umfassenden Analyse werden die Rotationsmodelle miteinander und mit einer reinen translatorischen Bewegung im Zusammenspiel mit den verschiedenen Enthakungsmethoden in Kapitel 5 verglichen.

### 4.3 Überwachte maschinelle Lernmethoden zur Auswahl geeigneter Enthakungspfadposen

Für die Auflösung von spezifischen Verhakungen wird eine situativ konkrete Strategie benötigt, welche ausgeführt wird und die möglichen Hemisphärenpfadposen auswählt, um mit höchster Wahrscheinlichkeit zum Erfolg zu kommen. Diese Strategie wird wegen ihrer Komplexität erlernt. In diesem Abschnitt werden dazu zwei überwachte, maschinelle Lernverfahren als Lösungsmethoden vorgestellt. Zu Beginn wird die Datengenerierung in einer Simulationsszene betrachtet. Durch diese Szene wird eine Datenbasis geschaffen, welche von beiden Lernmethoden verwendet werden kann. Nach der Datengenerierung werden zwei mögliche Anordnungen von neuronalen Netzwerken vorgestellt. Zu Beginn erfolgt eine Beschreibung einer Reihenschaltung von drei aufeinanderfolgenden, gefalteten, neuronalen Netzwerken sowie deren Interaktion. Im Anschluss wird eine aus den Reihenschaltungsnetzwerken zusammengeführte neuronale Gesamtarchitektur vorgestellt. Alle verwendeten, neuronalen Netzwerkarchitekturen sind gleich aufgebaut und werden am Ende dieses Abschnitts erläutert. Die Beschreibung des Trainings dieser Netzwerkarchitektur rundet diesen Abschnitt ab.

#### 4.3.1 Datengenerierung in einer Simulationsszene

Für das Training überwachter, maschineller Lernverfahren werden im Vorfeld Daten erzeugt oder gesammelt. Um Daten für die maschinellen Lernverfahren zur Auswahl geeigneter Enthakungspfadposen zu generieren, wird, wie bereits für die Verhakungserkennung in Abschnitt 3.1.3, eine Simulation genutzt. Der Ablauf und Algorithmus der Datengenerierung in der Simulationsszene wird in Abbildung 4.12 dargestellt. Für die Annotierung möglicher Hemisphärenpfadposen, die zu einer effizienten Enthakung einer verhakten Werkstücksituation in einer Kiste führen, werden CAD-Daten der Kiste und der Werkstücke benötigt. Diese werden in die Simulationsszene geladen. Zudem wird die Anzahl der in der Kiste befindlichen Werkstücke und die Anzahl von verhakten Werkstücksituationen definiert. Die verhakten Werkstücksituationen werden, wie bereits bei der Datengenerierung für die Verhakungserkennung in Abschnitt 3.1.3 beschrieben, über der Kiste gebildet und dann, bereits verhakt, randomisiert in die Kiste geworfen. Mögliche verhakte Lagen der Werkstücke

**Funktion: Annotierung für die Enthakung in der Simulation**

**Eingabe:** CAD-Daten der Kiste und Werkstücke, Anzahl  $n_W$  und Lage der Werkstücke mit  $n_V$  Verhakungen in der Kiste, Anzahl  $n_P$  und Lage der 17 Hemisphärenpfadposen und optionales Rotationsmodell, Anzahl  $n_K$  zu befüllender Kisten

**Ausgabe:** Annotierung der Hemisphärenpfadposen für eine bestmögliche Enthakung

```

0   $n = 0$ 
1  solange  $n < n_K$  tue
2      fülle Kiste mit  $n_W$  Werkstücken darin  $n_V$  Verhakungen (Ausgangssituation)
3      für jede Verhakung tue
4          wähle randomisierte Greifpose am verhakten Werkstück aus
5          erstelle erste Hemisphäre  $H1$  mit  $n_P = 17$  Hemisphärenpfadposen
6          für jede Hemisphärenpfadpose  $P_{H1,i}$  der ersten Hemisphäre tue
7               $P_{H1,i}$  anfahen, bewerten, danach Ausgangssituation herstellen
8              beste Hemisphärenpfadpose  $P_{H1}^*$  auf erfolgreiche Enthakung prüfen
9              wenn Enthakung nicht erfolgreich dann
10                 stelle Ausgangssituation her
11                 erstelle  $H2$  mit Ursprung in  $P_{H1}^*$ 
12                 für jede  $P_{H2,i}$  tue
13                      $P_{H1}^*$  anfahen
14                      $P_{H2,i}$  anfahen, bewerten, danach Ausgangssituation herstellen
15                      $P_{H2}^*$  auf erfolgreiche Enthakung prüfen
16                     wenn Enthakung nicht erfolgreich dann
17                         stelle Ausgangssituation her
18                         erstelle  $H3$  mit Ursprung in  $P_{H2}^*$ 
19                         für jede  $P_{H3,i}$  tue
20                              $P_{H1}^*$  und  $P_{H2}^*$  anfahen
21                              $P_{H3,i}$  anfahen, bewerten, Ausgangssituation herstellen
22                              $P_{H3}^*$  auf erfolgreiche Enthakung prüfen
23                             wenn Enthakung nicht erfolgreich dann
24                                 bewerte alle Pfadposen aller Hemisphären als ungenügend
25                                 stelle Ausgangssituation her
26                                 sonst bewerte  $P_{H3}^*$  am besten und Ausgangssituation
27                                 sonst bewerte  $P_{H3,17}$  am besten
28                                 stelle Ausgangssituation her
29                                 sonst bewerte  $P_{H2,17}$  und  $P_{H3,17}$  am besten
30                                 stelle Ausgangssituation her
31          Kiste für erneute Befüllung leeren
32       $n = n + 1$ 
33 Ende

```

Abbildung 4.12: Simulationsablauf zur Datengenerierung für die Enthakungsmethode

werden im Vorfeld in einer Simulationsszene bestimmt und geladen. Neben den Werkstücken und der Kiste werden die Anzahl an Hemisphärenpfadposen für jede Hemisphäre sowie deren Lage benötigt. Zudem ist es möglich, optional die Orientierungsänderungen zuzuschalten. Die Menge an benötigten Daten wird über die Anzahl der zu befüllenden Kisten bestimmt. Durch die Datengenerierung wird ein Datensatz erstellt, der für einen Tiefenkartenzuschnitt einer verhakten Werkstücksituation drei Hemisphärenpfadposen in drei Hemisphären beinhaltet, welche zu einer erfolgreichen Enthakung führen können.

Solange die Anzahl an zu befüllenden Kisten noch nicht erreicht ist, werden zu Beginn der Simulationsszene Werkstücke in eine Kiste abgeworfen. Eine befüllte Kiste mit  $n_W$  Werkstücken mit  $n_V$  Verhakungen wird als Ausgangssituation bezeichnet.

Für jede Verhakung in der Kiste wird zuerst ein Tiefenkartenzuschnitt erstellt und eine Greifpose aus den vordefinierten Möglichkeiten, beschrieben in Abschnitt 4.1.2, randomisiert ausgewählt. Diese wird zunächst auf Kollisionen mit der Punktwolke des Tiefenbildes überprüft. Sofern eine kollisionsfreie Greifpose gefunden wurde, wird diese für die verhakte Werkstücksituation beibehalten. Über das ausgewählte, verhakte Werkstück wird eine Hemisphäre mit  $n_P$  Hemisphärenpfadposen gespannt. Hierbei ist die Orientierung, wie in Abschnitt 4.2.3 beschrieben, durch die  $x$ -Achse des Werkstückkoordinatensystems  $K_W$  und die Kistenbodenebene gegeben. Im Anschluss wird jede der  $n_P$  Hemisphärenpfadposen angefahren und nach der Funktion aus Abschnitt 4.2.2 bewertet. Die Hemisphärenpfadpose mit der besten Bewertung wird dann nochmals angefahren und wie in Abschnitt 3.1.3 auf Enthakung überprüft. Sollte durch die Bewegung zu einer Hemisphärenpfadpose eine Kollision des Greifers mit dem gegriffenen Werkstück auftreten, wird diese als unmöglich annotiert.

Wenn die Enthakung nach der ersten Bewegung erfolgt ist, werden in der zweiten und dritten Hemisphäre jeweils die Hemisphärenpfadposen mit der Nummer 17 als am besten deklariert. Die Hemisphärenpfadposen mit der Nummer 17 befinden sich senkrecht zur Kistenbodenebene direkt über dem verhakten Werkstück. Durch diese Annotierung kann später ein Abbruch der Enthakungsbewegung erkannt und der Entnahmepfad ohne weitere,



zusätzliche Pfadposen fortgeführt werden. Wenn die Enthakung nach der ersten Bewegung noch nicht erfolgt ist, wird die am besten bewertete Hemisphärenpfadpose aus der ersten Hemisphäre als Ursprung für die zweite Hemisphäre angenommen und die zweite Hemisphäre aufgespannt. Die zweite Hemisphäre wird hierbei gleich wie die erste Hemisphäre orientiert. Zudem wird die Ausgangssituation wieder hergestellt.

Nun werden nacheinander die Hemisphärenpfadposen der zweiten Hemisphäre angefahren und bewertet. Jede Bewegung besteht dabei zuerst aus einer Bewegung zu der ausgewählten Pfadpose in der ersten Hemisphäre und der jeweiligen Pfadpose aus der zweiten Hemisphäre. Nach der abgeschlossenen Bewegung zu jeder Hemisphärenpfadpose wird die am besten bewertete wieder ausgewählt und auf Enthakung geprüft. Sollte die Enthakungsprüfung nach zwei zusätzlichen Pfadposen einen Erfolg aufweisen, wird die Pfadpose mit der Nummer 17 in der dritten Hemisphäre am besten bewertet. Sollte die Enthakungsprüfung nicht erfolgreich, die Verhakung also immer noch präsent sein, wird der Prozess für eine dritte Hemisphäre wiederholt. Wenn eine Enthakung auch nach drei zusätzlichen Hemisphärenpfadposen und damit nach drei aufgespannten Hemisphären nicht gelöst sein sollte, werden alle Pfadposen als ungenügend  $c_{n_P} = 1$  annotiert. Wenn die Enthakung nach drei Hemisphären gelöst wurde, wird die ausgewählte Hemisphärenpfadpose aus der dritten Hemisphäre am besten bewertet  $c_{n_P} = 0$  und die Simulationsszene für eine erneute Kistenbefüllung vorbereitet.

Als Ausgabe entstehen somit Bewertungen für alle 17 Hemisphärenpfadposen der ersten, zweiten und dritten Hemisphäre, welche gemeinsam mit dem zugehörigen Tiefenkartenschnitt und der ausgewählten Greifpose in einem Datensatz gespeichert werden.

### **4.3.2 Enthakungspfadplanung durch eine Reihenschaltung neuronaler Netzwerkarchitekturen**

Die erste überwachte maschinelle Lernmethode zur Enthakungspfadplanung besteht aus drei gefalteten, neuronalen Netzwerken (CNN), die in Reihe geschaltet sind. Abbildung 4.13 stellt die Reihenschaltung und die Interaktion schematisch dar. Der Tiefenkartenschnitt der verhakten Werkstücksituation und die Greifpose als Transformationsmatrix

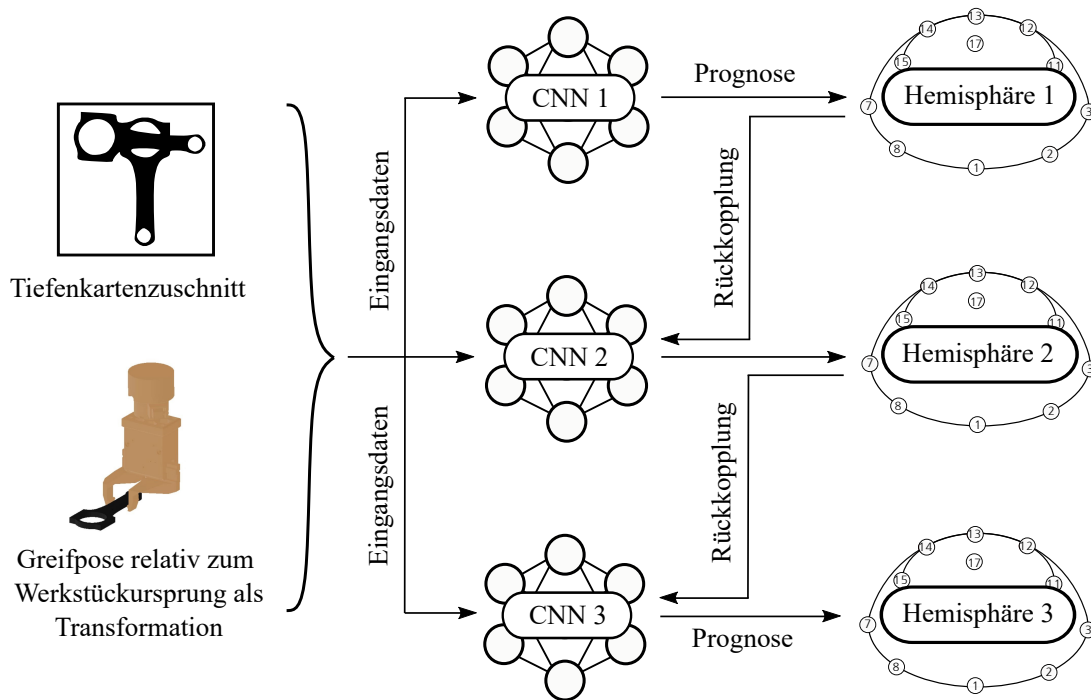


Abbildung 4.13: Reihenschaltung von drei neuronalen Netzwerken zur Enthakungspfadplanung und deren Interaktion in Anlehnung an Moosmann et al. (2021b)

${}^T_f T_W$  relativ zum Werkstückursprung  $K_W$  beschreiben die Eingangsdaten in die Reihenschaltung und sind für alle drei Netzwerkarchitekturen (CNN 1, CNN 2, CNN 3) gleich. Die Netzwerkarchitekturen sind bis auf veränderliche Eingangsdaten gleich, werden auch für die zusammengeführte Netzwerkarchitektur in Abschnitt 4.3.3 verwendet und in Abschnitt 4.3.4 beschrieben.

Der Prozess beginnt mit dem CNN 1. Nachdem die Eingangsdaten in das CNN gespeist wurden, werden 17 Werte prognostiziert. Die Werte werden im Intervall  $[0,1]$  ausgegeben und beschreiben für jede Hemisphärenpfadpose eine Schätzung für die Eignung zur Enthakung der vorliegenden, verhakten Werkstücksituation. Die beste Prognose, also die beste ausgewählte Hemisphärenpfadpose in der ersten Hemisphäre, wird nun durch die Greifplanung auf mögliche Kollisionen überprüft. Sollte die prognostizierte Hemisphärenpfadpose nicht anfahrbar sein, so wird die nächstbeste Prognose ausgewählt und ebenfalls auf Kollisionen überprüft. Dieser Prozess wird so lange wiederholt, bis eine bestmögliche, kollisionsfreie Pfadpose gefunden wurde. Der Index dieser Hemisphärenpfadpose wird in Form einer Matrix  $M_a$  als Rückkopplung an das CNN 2 übergeben. Die  $17 \times 2$  Matrix

beinhaltet in der ersten Spalte 17 Skalare für die 17 möglichen Hemisphärenpfadposen der ersten Hemisphäre sowie 17 Skalare in der zweiten Spalte für Hemisphäre 2. Nach der ersten ausgewählten Pfadpose wird die erste Spalte der Matrix befüllt. Der Eintrag der ausgewählten Pfadpose beträgt 1, alle anderen Pfadposen werden mit 0 eingetragen. Die zweite Spalte der Matrix wird für die erste Rückkopplung nicht benötigt. Mit der beschriebenen Matrix und den bereits bekannten Eingangsdaten des CNN 1, prognostiziert das CNN 2 17 Werte für die 17 möglichen Hemisphärenpfadposen in der zweiten Hemisphäre. Nach Kollisionsprüfung und Auswahl einer geeigneten und anfahrbaren Pfadpose wird die Matrix  $M_a$  in der zweiten Spalte durch die ausgewählte Pfadpose ergänzt und dient als Eingangsdaten für das CNN 3. Zudem werden wieder der Tiefenkartenzuschnitt der verhakten Werkstücksituation und die Greifpose relativ zum Werkstückursprung zur Verfügung gestellt. Das CNN 3 prognostiziert, wie bereits CNN 1 und CNN 2, die Eignung der 17 Hemisphärenpfadposen für eine mögliche Enthakung. Sobald die dritte Pfadpose auf Kollisionen überprüft und ausgewählt wurde, endet die Enthakungspfadplanung und die Pfadposen werden an die Robotersteuerung übergeben.

### 4.3.3 Enthakungspfadplanung durch eine zusammengeführte, neuronale Gesamtarchitektur

Die zweite überwachte maschinelle Lernmethode zur Enthakungspfadplanung vereint die in Abschnitt 4.3.2 vorgestellten Netzwerke in einem gefalteten, neuronalen Netzwerk. Während in der Reihenschaltung die neuronalen Netzwerke unterschiedliche Gewichte beinhalten, werden in dem zusammengeführten neuronalen Netzwerk Gewichte zur Prognose für alle drei Hemisphären gebildet. In Abbildung 4.14 sind die zusammengeführte neuronale Gesamtarchitektur und die Interaktion abgebildet.

Der Prozess beginnt mit der Einspeisung der Eingangsdaten in das zusammengeführte Netzwerk. Neben dem Tiefenkartenzuschnitt der verhakten Werkstücksituation werden die Greifpose als Transformationsmatrix  ${}^fT_W$  relativ zum Werkstückursprung  $K_W$  und eine  $17 \times 2$  Matrix in das Netzwerk gegeben. Die Matrix wird, wie bereits in Abschnitt 4.3.2, nach jeder Hemisphäre mit dem Index der ausgewählten Hemisphärenpfadpose befüllt.

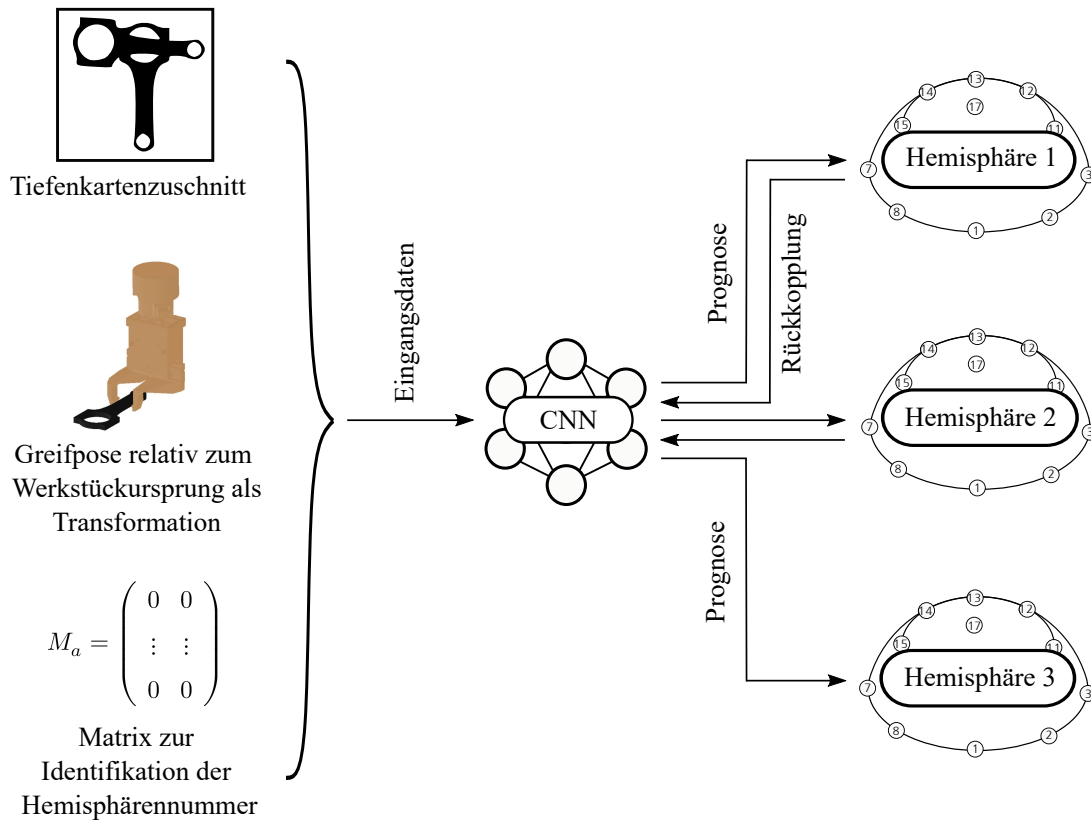


Abbildung 4.14: Zusammengeführte, neuronale Netzwerkarchitektur zur Enthakungspfadplanung und deren Interaktion

Zu Beginn, also vor der Prognose für die erste Hemisphäre, ist diese Matrix nur mit Nullen befüllt. Nach der Prognose der ersten möglichen Hemisphärenpfadpose und der Überprüfung auf Kollisionen, wird das gleiche Netzwerk mit einer angepassten Matrix  $M_a$  ein zweites Mal zur Bewertung der 17 Hemisphärenpfadposen der zweiten Hemisphäre beansprucht. Der Prozess endet mit der Auswahl der besten und kollisionsfrei anfahrbaren Hemisphärenpfadpose und die Übergabe der drei Posen an die Robotersteuerung.

#### 4.3.4 Netzwerkarchitektur zur Enthakungspfadplanung durch überwachte Lernmethoden

Die vorgestellte Netzwerkarchitektur ähnelt einem DenseNet und wird für beide überwachte Lernmethoden zur Enthakungspfadplanung verwendet. Die erste Verwendung eines sogenannten DenseNets wurde von Huang et al. (2017) beschrieben. Der Einsatz eines DenseNets bringt viele Vorteile mit sich. Die Architekturen ermöglichen eine effiziente Nutzung von Ressourcen wie Speicher und Rechenleistung. Durch die Verbindung von

Schichten in einem dichten Netzwerk können Parameter besser wiederverwendet werden, was zu einer geringeren Anzahl insgesamt benötigter Parameter führt. Dadurch kann die DenseNet-Architektur mit weniger Ressourcen auskommen und dennoch gute Leistung erzielen. Zudem ermöglicht die Verbindung jeder Schicht mit allen nachfolgenden Schichten einen direkten Gradientenfluss. Dadurch wird das Problem des Verschwindens des Gradienten reduziert, das bei sehr tiefen neuronalen Netzwerken auftreten kann. Durch die enge Verbindung der Schichten wird zudem ein direkter Austausch von Informationen zwischen verschiedenen Ebenen des Netzwerks ermöglicht. Jede Schicht erhält Eingaben von allen vorherigen Schichten und gibt ihre Ausgabe an alle nachfolgenden Schichten weiter. Dadurch wird das Modell dazu angeregt, aus den gesamten bisherigen Schichten zu lernen, anstatt nur aus den unmittelbar vorherigen Schichten. Dieser Wissensaustausch kann zu einer besseren Modellgenauigkeit führen. Ein Denseblock besteht dabei aus mehreren Denseschichten, wobei jede Schicht jeweils alle Ausgaben der vorherigen Schichten als Eingabe erhält. In die Eingangsschicht wird der Tiefenkartenzuschnitt der verhakten Werkstücksituation in Form eines  $128 \times 128$  Pixel großen Bildes eingegeben. Auf die Eingangsschicht folgen vier Denseblöcke mit den Größen  $64 \times 64 \times 10$ ,  $32 \times 32 \times 58$ ,  $16 \times 16 \times 154$  und  $8 \times 8 \times 346$ . Die Denseschichten bestehen jeweils aus einer Normalisierungsschicht der Datenstapel, einer ReLU-Aktivierungsschicht und einer Faltungsschicht. Jede dieser Denseschichten erstellt, mit seinen Filtern und allen bis dahin zur Verfügung stehenden Informationen, Merkmalskarten, die dem Informationspool der Netzwerkarchitektur wiederum für alle weiteren Schichten zur Verfügung gestellt werden. Da diese ständige Wiederverwendung der Merkmalskarten nur im Falle von gleichen Merkmalskartengrößen möglich ist, wird das Netzwerk in die Denseblöcke unterteilt, innerhalb welchen die Auflösung der Ein- und Ausgaben der Filter konstant gehalten wird. Die für die Merkmalsextraktion eines CNN elementare Herunterskalierung wird zwischen den Denseblöcken durch die Transitionsschichten realisiert. Diese verfügen neben einer Normalisierung der Datenstapel und einer ReLU-Aktivierung auch über eine Durchschnittsbündelung (engl. *Average Pooling*). Bei dieser werden die Merkmalskarten für den nächsten Denseblock anhand von Filtern mit der Größe  $2 \times 2$  und einer Schrittweite von 2 in ihrer Größe halbiert. Auf diese Weise durchlaufen die Tiefenkarten die Netzwerkarchitektur, bis nach dem letzten Denseblock alle Merkmale anhand einer Durchschnittsbündelung zusammengefasst und mit einem

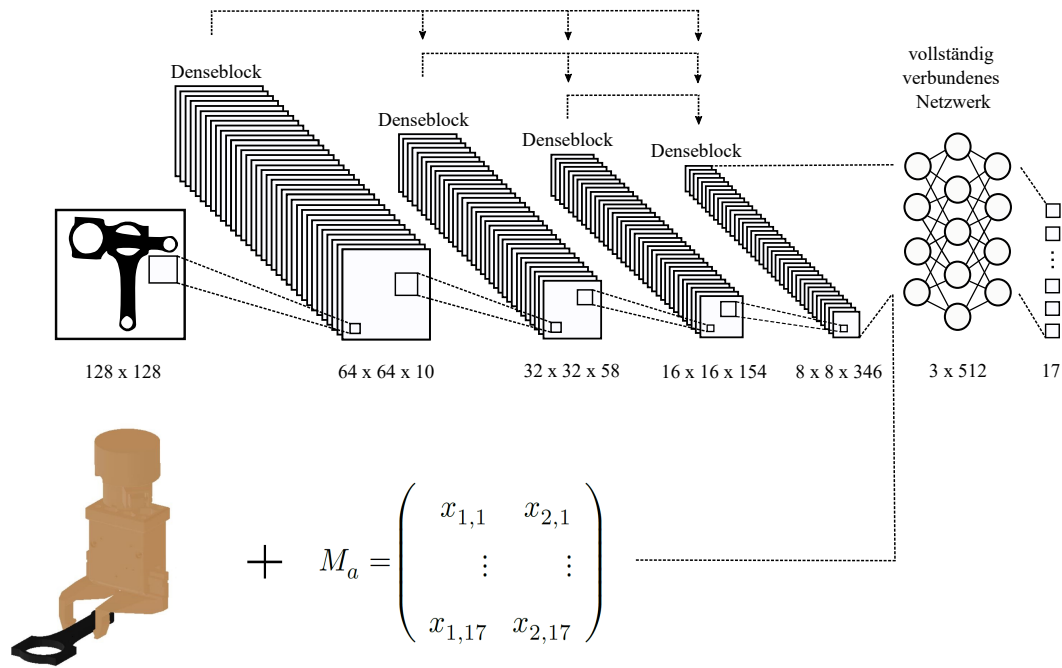


Abbildung 4.15: Gefaltete, neuronale Netzwerkarchitektur der überwachten Lernmethoden zur Enthakungspfadplanung in Anlehnung an Moosmann et al. (2021b)

vollständig verbundenen Netzwerk auf beliebig viele Ausgaben verknüpft werden. Um den generellen Aufbau des Netzwerks zu vervollständigen, befindet sich vor dem ersten Denseblock, direkt nach der Eingabe, bereits eine erste initiale Faltungsschicht. Vor dem vollständig verbundenen Netzwerk aus drei Schichten mit jeweils 512 Neuronen werden die Transformationsmatrix  ${}^T_f T_W$  der Greifpose relativ zum Werkstückursprung  $K_W$  und die Matrix  $M_a$  der vorherigen Entscheidungen eingespeist.

Die vorgestellte Netzwerkarchitektur ist in Abbildung 4.15 dargestellt. Das Training des gefalteten, neuronalen Netzwerks unterscheidet sich zwischen den zwei vorgestellten Lernmethoden. Für die Reihenschaltung wird jedes CNN einzeln trainiert. Die Daten werden, wie in Abschnitt 4.3.1 beschrieben, in einer Simulationsszene generiert und können für jede Hemisphäre einzeln abgerufen werden. Das Training der Lernmethode mit einer zusammengeführten, neuronalen Netzwerkarchitektur wird für jede verhakte Werkstücksituation mit den Einträgen in der Matrix  $M_a$  durchlaufen.

Die neuronale Netzwerkarchitektur, welche für beide Lernmethoden eingesetzt und zuvor beschrieben wurde, wird mit 20.000 Datenbeispielen in einem Datensatz trainiert. Ein

Datenbeispiel besteht hierbei aus einem Tiefenkartenzuschnitt einer verhakten Werkstücksituation, einer Greifpose in Form einer Transformationsmatrix  ${}^T T_W$  relativ zum Werkstückkoordinatensystem  $K_W$ , der Matrix  $M_a$  zur Identifikation der Hemisphärenindizes und 17 bewerteten Hemisphärenpfadposen in Form eines Vektors mit 17 normalisierten Einträgen im Intervall  $[0,1]$ . Zur Validierung werden 2.000 und zum Test des Trainings 4.000 Datenbeispiele verwendet. Jedes Netzwerk wird in 50 Epochen mit einer Lernrate von  $2 \cdot 10^{-4}$  trainiert. Die Stapelgröße der Datenbeispiele beträgt für alle CNN der Reihenschaltung 128. Die Verlustfunktion, die das Training steuert, ist wie bereits für die Verhakungserkennung in Abschnitt 3.3 beschrieben, eine binäre Kreuzentropie-Verlustfunktion. Der Verlust für das Training der neuronalen Netzwerkarchitektur wird mittels

$$L_{Ent} = \sum_{i=1}^{n_V} \sum_{j=1}^{n_H} \sum_{k=1}^{n_P} (l_{V_i, H_j, P_k} \cdot \log(s_{V_i, H_j, P_k}) + (1 - l_{V_i, H_j, P_k}) \cdot \log(1 - s_{V_i, H_j, P_k})) \quad (4.22)$$

berechnet. Hierbei wird nicht nur über die Anzahl  $n_V$  der verhakten Werkstücksituationen iteriert, sondern auch über die Anzahl  $n_H$  an Hemisphären und die Anzahl  $n_P$  an Hemisphärenpfadposen mit den jeweiligen Annotierungen  $l_{V_i, H_j, P_k}$  und Prognosen  $s_{V_i, H_j, P_k}$ . Die Ergebnisse des Trainings und der Verhakungserkennung werden in Kapitel 5 vorgestellt.

#### 4.4 Bestärkende maschinelle Lernmethode zur Auswahl geeigneter Enthakungspfadposen

Neben den bereits vorgestellten überwachten Lernverfahren zur Auswahl geeigneter Enthakungspfadposen wird in diesem Abschnitt eine Methode des bestärkenden Lernens vorgestellt. Die Methode basiert ebenfalls auf dem Hemisphärenmodell und der Bewertungsfunktionen der Hemisphärenpfadposen. Im Folgenden wird die bestärkende Lernmethode für die Enthakungspfadplanung entwickelt und deren Training erläutert.

##### 4.4.1 Training eines Agenten für das bestärkende Lernen

Im Gegensatz zu den überwachten Lernverfahren, für deren Training zuerst Daten überwacht generiert und im Nachgang künstliche, neuronale Netzwerke trainiert werden, lernt

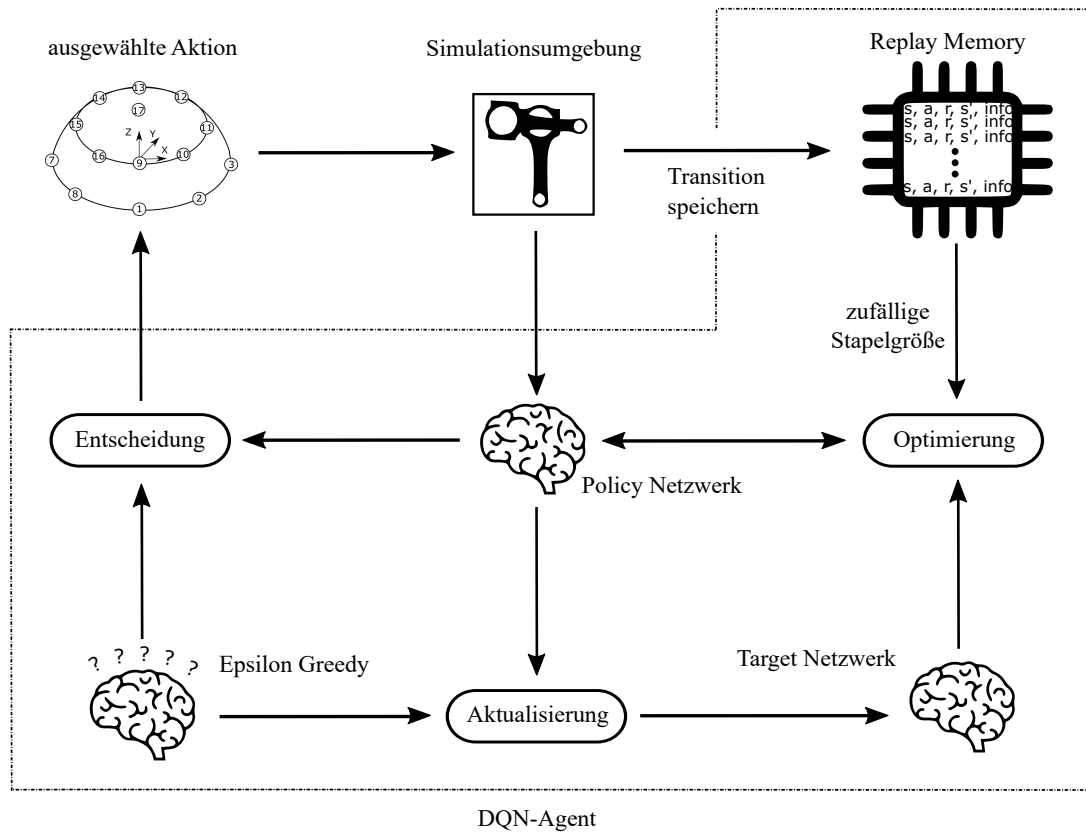


Abbildung 4.16: Trainingsumgebung, Bestandteile und Interaktion des Agenten für das bestärkende Lernen zur Enthakungspfadplanung (Moosmann et al. 2021a)

das bestärkende Lernen durch Belohnung und Bestrafung. Hierfür wird ebenfalls eine Simulationsszene initialisiert und über eine Schnittstelle ein paralleles Netzwerktraining hergestellt. In der Simulationsszene läuft für eine verhakten Werkstücksituation der gleiche Prozess wie bei der Datengenerierung für die überwachten Lernverfahren ab. Es wird eine Kiste mit  $n_W$  Werkstücken und  $n_V$  Verhakungen befüllt. Eine verhakten Werkstücksituation wird ausgewählt, der Tiefenkartenzuschnitt angefertigt und eine erste Hemisphäre über die Situation gespannt. Der Agent für das bestärkende Lernen wählt dann eine Hemisphärenpfadpose aus und löst damit das Erzeugen einer zweiten Hemisphäre aus. Zur Erzeugung einer dritten Hemisphäre wird eine Hemisphärenpfadpose auf der zweiten Hemisphäre vom Agenten ausgewählt. Der Agent wählt anschließend eine Hemisphärenpfadpose auf der dritten Hemisphäre aus. Die Trainingsumgebung sowie die Bestandteile des Agenten für das bestärkende Lernverfahren sind in Abbildung 4.16 dargestellt.



Der Agent besteht aus einem *Replay Memory*, einem *Policy Netzwerk*, einem *Target Netzwerk* und verfolgt die *Epsilon Greedy Strategie*. Bei dem *Deep Q-Learning-Agent*, kurz DQN-Agent, handelt es sich um eine wertebasierte (engl. *value based*), modellfreie *Off-Policy Methode*. Ein modellfreier Ansatz bedeutet, dass die Umgebung zwar bekannt ist, allerdings nicht in welchem Zustand durch welche Aktion, welche Belohnung hervorgerufen wird. Die Wertebasiertheit des Ansatzes folgt der Annahme, dass für jede Aktion, die der Agent vorgibt, eine Bewertung durch eine Nutzenfunktion erstellt werden kann.

Die *Off-Policy Methode* bedeutet, dass der DQN-Agent nicht nur die von einer Strategie ausgewählten Aktionen lernen kann, sondern darüber hinaus über die Möglichkeit verfügt, durch weitere Strategien, Erfahrungen zu sammeln. Eine dieser Erkundungsstrategien ist beispielsweise die *Epsilon Greedy Strategie*. Aktionen werden dabei mit einer Wahrscheinlichkeit  $\epsilon$  zufällig gewählt. Das  $\epsilon$  wird im Laufe des Trainings von  $\epsilon_{max} = 1,0$  mit einem Abschwächungsfaktor  $\epsilon_{step} = 10^{-4}$  weiter reduziert bis zu einer Untergrenze  $\epsilon_{min} = 10^{-3}$ , sodass immer weniger zufällige Aktionen ausgewählt werden (Vermorel & Mohri 2005). Aktualisierungen werden gemäß der *Temporal Difference Methode* berechnet (Watkins 1989), was bedeutet, dass nicht nach jeder Episode aktualisiert wird, wie beispielsweise bei der Monte Carlo Methode (Sutton & Barto 2018), sondern nach einem bestimmten Zeitschritt. Der Zeitschritt wird so gewählt, dass Aktualisierungen nach jeder Hemisphärenpfadpose erfolgen können.

Eine *Off-Policy Methode* weist häufig Instabilität und Divergenz im Training auf. Deshalb wird ein *Replay Memory* installiert. Im *Replay Memory* werden alle von dem Agenten gemachten Erfahrungen gespeichert, welche während des Trainings dem Wiederholen von Erfahrungen (engl. *Experience Replay*) dient. Die gespeicherten Erfahrungen bestehen aus dem aktuellen Zustand  $s_t$ , der ausgewählten Aktion  $a_t$ , der damit verbundenen Belohnung  $r_t$ , dem Folgezustand  $s_{t+1}$ , welcher erreicht wird, wenn im Zustand  $s_t$  die Aktion  $a_t$  ausgeführt wurde, und einer Information *info*, welche beschreibt, ob diese Aktion eine finale Aktion ist, also eine Bewegung auf der dritten Hemisphäre. Der Zustand  $s_t$  wird dabei mit den gleichen Eingangsdaten wie beim überwachten Lernverfahren de-

finiert. Er beinhaltet den Tiefenkartenzuschnitt und die Greifpose. Die Aktion  $a_t$  ist durch die ausgewählte Hemisphärenpfadpose definiert. Die gespeicherten Erfahrungen werden Übergangsinformation (engl. *Transition*) genannt. Diese Übergangsinformationen werden in jedem Zeitschritt im *Replay Memory* gespeichert. Die Information über einen finalen Zustand ist für die Berechnung durch die *Temporal Difference Methode* entscheidend. Dadurch, dass in jedem Zeitschritt eine Auswahl mehrerer zufällig ausgewählter Übergangsinformationen verwendet werden, um die Gewichte des *Policy Netzwerks* zu aktualisieren, werden Erfahrungen in der Regel mehrfach zum Trainieren verwendet. Die Dateneffizienz des DQN-Agenten wird hierdurch gesteigert (Mnih et al. 2015). Außerdem verhindert die Nutzung vergangener Erfahrungen das Oszillieren und Divergieren der Parameter (Mnih et al. 2013). Die Verwendung des Wiederholens von Erfahrungen beeinflusst die Auswahl des Optimierungsverfahrens des *Policy Netzwerks*. *RMSProp* ist eine Modifikation der Werterückführung (engl. *Back-Propagation*), welche sich in diesem Kontext als besonders geeignet darstellt (Mnih et al. 2015) und an dieser Stelle verwendet wird.

Das *Q-Learning* wird eingesetzt, um den Aktionswert

$$Q(s_t, a_t) \leftarrow (1 - \alpha)Q(s_t, a_t) + \alpha[r_{t+1} + \gamma \max_a Q(s_{t+1}, a)] \quad (4.23)$$

zu aktualisieren. Die Aktualisierung (Watkins 1989) wird mit dem Diskontierungsfaktor  $\gamma = 0,99$  und der Lernrate  $\alpha = 3 \cdot 10^{-3}$  berechnet.

Das *Target Netzwerk* dient ebenfalls der Stabilität des Trainings (Lin 1993; Mnih et al. 2015; François-Lavet et al. 2018) und wird in bestimmten Zeitschritten durch das *Policy Netzwerk* aktualisiert. Die Einführung dieses zweiten neuronalen Netzwerks hat sich bereits bei dem Training von Atari-Spielen als sehr brauchbar herausgestellt. Die Optimierungen, die das *Target Netzwerk* dem DQN-Agenten zur Verfügung stellt, erzeugen neben dem *Replay Memory*, die größte Stabilitätskomponente des Agententrainings. Hierbei ist das *Target Netzwerk* eine Kopie des *Policy Netzwerks* und wird in diesen Zeitintervallen überschrieben. Das *Policy Netzwerk* ist schlussendlich das neuronale Netzwerk, welches nach dem Training für die Prognosen zur Auswahl der Hemisphärenpfadposen verwendet wird.

Die Architekturen des *Target Netzwerks* und des *Policy Netzwerks* werden in Abschnitt 4.4.2 beschrieben.

Die Interaktion des DQN-Agenten mit der Umgebung und die Optimierung des *Policy Netzwerks* sind in Abbildung 4.16 visualisiert. Der DQN-Agent erhält zwei Eingangswerte aus der Umgebung. Dies ist der Zustand, welcher direkt an das *Policy Netzwerk* gesendet wird. Die Transition wird direkt an das *Replay Memory* gesendet. Als Ausgabe des DQN-Agenten ist die gewählte Aktion zu sehen, welche den einzigen Eingangswert für die Umgebung darstellt. Dies entspricht der prinzipiellen Interaktion zwischen Umwelt und Agenten, also der Auswahl einer Hemisphärenpfadpose. Eine detaillierte Betrachtung des DQN-Agenten zeigt, dass die Aktion entweder zufällig über die *Epsilon Greedy Strategie* oder durch die Vorhersage des *Policy Netzwerks* ausgewählt werden kann. Sobald eine Aktion ausgewählt wurde und die Umwelt vom Zustand  $s_t$  in den Folgezustand  $s_{t+1}$  übergeht, wird die Transition im *Replay Memory* gespeichert. Gleichzeitig wird der Folgezustand  $s_{t+1}$  dem *Policy Netzwerk* als neuer Zustand  $s_t$  übergeben. Um das *Policy Netzwerk* zu trainieren, werden die gespeicherten Übergangsinformationen als Zufallsstapel (engl. *Random Batch*) aus dem *Replay Memory* entnommen. Dabei beinhaltet der Stapel eine randomisiert ausgewählte Stapelgröße. Aus diesen Übergangsinformationen wird die Verlustfunktion, die mittlere quadratische Abweichung, berechnet. Dazu berechnet das *Policy Netzwerk* den Q-Wert  $Q(s_t, a_t)$  des Zustandes  $s_t$  und das *Target Netzwerk* den Q-Wert  $Q(s_{t+1}, a_{t+1})$  des Folgezustandes  $s_{t+1}$ . Das *Target Netzwerk* wird alle 1.000 Zeitschritte durch das *Policy Netzwerk* aktualisiert.

Sobald das *Policy Netzwerk* trainiert ist und die Lösungen des Netzwerks konvergieren, wird der Testprozess gestartet. Für den Testprozess werden das *Policy Netzwerk*, die ausgewählte Aktion  $a_t$  und die Umgebung verwendet. Bei der Aktionswahl wird die *Epsilon Greedy Strategie* nicht mehr mit einbezogen, da nun keine weitere Erkundung der Umgebung erwünscht ist. Im Testprozess erhält das *Policy Netzwerk* den Zustand  $s_t$  aus der Umwelt und bestimmt auf dieser Basis eine geeignete Aktion  $a_t$ , welche direkt an die Umwelt weitergeleitet wird und zu dem Folgezustand  $s_{t+1}$  führt, welcher wiederum den neuen Eingabewert  $s_t$  für das *Policy Netzwerk* bildet.

Die Belohnungsstrategie (engl. *Reward Shaping*) wird durch die Bewertungsfunktion der Hemisphärenpfadposen aus Abschnitt 4.2.2 abgebildet. Somit bleibt das bestärkende Lernverfahren mit den überwachten Lernverfahren vergleichbar. Für jede ausgewählte Aktion wird der Faktor der Unruhe  $d_{W_i}^n$  und der Faktor des Enthakungsfortschritts  $d_{W_{i,v}}^n$  mit dem Gewichtungsfaktor  $w_P$  zu den Kosten einer Hemisphärenpfadpose  $c_{n_P}$  berechnet. Um eine Enthakung der verhakten Werkstücksituation zu belohnen, wird ein zusätzlicher Parameter  $c_U$  zu der Gesamtbelohnung addiert. Der Faktor  $c_U = 100$  wurde empirisch ermittelt. Die Berechnung der Belohnung für eine ausgewählte Aktion wird mittels

$$R(a_t) = (1 - c_{n_P}) + c_U \quad (4.24)$$

berechnet. Der Grund für die Einführung eines zusätzlichen Faktors besteht in der Forcierung des Trainings hinsichtlich einer erfolgreichen Enthakung und bietet die Möglichkeit, im späteren Verlauf dieser Arbeit, auf situativ unmögliche Enthakungsstrategien, wie beispielsweise Greifposen, die eine Enthakung blockieren, zu reagieren.

Um eine ausreichende Konvergenz des Trainings des bestärkenden Lernverfahrens zu kontrollieren und das Training daraufhin zu beenden, wird ein Konvergenzkriterium eingeführt. Hierfür werden 100 repräsentative, verhakte Werkstücksituationen manuell ausgewählt und jeweils nach 100 Episoden alle 100 Situationen durch Prognosen des *Policy Netzwerks* auf Enthakung geprüft. Wird eine bestimmte Anzahl an Enthakungen erreicht, wird das Training abgebrochen. Diese Anzahl wird empirisch ermittelt. In der Entwicklung dieser Arbeit wird eine Enthakungsrate von 95% verwendet. Dies bedeutet, dass von 100 repräsentativen Verhakungen, 95 durch den aktuellen Trainingsstand enthakt wurden und das Training daraufhin beendet wird.

#### 4.4.2 Netzwerkarchitektur des bestärkenden Lernverfahrens

Die Netzwerkarchitektur des *Policy Netzwerks* sowie des *Target Netzwerks* besteht aus mehreren Schichten. Sie ist in Abbildung 4.17 dargestellt. Der Einsatz dieser neuronalen Netzwerkarchitektur ist durch viele Faktoren beeinflusst. Durch die Anwendung von Faltungsschichten mit verschiedenen Filtergrößen wird eine hierarchische Merkmalsextraktion

ermöglicht. Dies hilft dabei, relevante Merkmale und Muster in den Eingabedaten zu identifizieren und zu erfassen. In diesem Fall werden Merkmalskarten mit verschiedenen Größen erzeugt, um Informationen in verschiedenen Abstraktionsebenen zu repräsentieren. Dies ermöglicht zudem eine Reduktion der Dimensionalität der Daten und hilft dabei, unnötige Informationen zu entfernen und gleichzeitig wichtige Merkmale beizubehalten. Eine niedrigere Dimensionalität erleichtert auch das Training der nachfolgenden vollständig verbundenen Schichten. Die Skalierbarkeit des Netzwerks ermöglicht zudem, die Kapazität zu erhöhen und komplexere Zusammenhänge zwischen den Eingabedaten und den Aktionsentscheidungen zu erfassen. Zu Beginn wird der Tiefenkartenzuschnitt der verhakten Werkstücksituation mit der Bildgröße  $128 \times 128$  in die Eingangsschicht des Netzwerks eingespeist. Durch Faltungen mit der Filtergröße  $8 \times 8$  und einer Schrittgröße von 4 und anschließendem *Max Pooling*, entstehen 32 Merkmalskarten mit der Größe  $32 \times 32$  Pixel. Durch weitere Faltungen mit Filtergrößen von  $4 \times 4$  Pixel und anschließendem *Max Pooling*, wird die Größe der Merkmalskarten halbiert und die Anzahl der Merkmalskarten auf 64 in der dritten Faltungsschicht verdoppelt. Nach den Faltungsschichten folgt ein vollständig verbundenes Netzwerk mit drei Schichten und jeweils 512 Neuronen. Die Transformationsmatrix  ${}^T_f T_W$  sowie die Matrix  $M_a$  vergangener, ausgewählter Hemisphärenpfadposen werden als zusätzliche Eingangswerte in das vollständig verbundene Netzwerk eingespeist. Als Ausgabe werden 17 Werte für die Hemisphärenpfadposen prognostiziert. Als Aktivierungsfunktion wird in allen Schichten die ReLU-Funktion verwendet.

## 4.5 Transfer von der Simulation in die reale Roboterzelle

Die zwei vorgestellten, überwachten Lernverfahren werden mit Daten trainiert, die in einer Simulationsumgebung entstanden sind. Gleiches gilt für das bestärkende Lernverfahren. Hier wird ein DQN-Agent in einer Simulationsumgebung auf Enthakungen trainiert. Simulationsdaten werden, wie bereits in Abschnitt 3.2 beschrieben, unter idealen Bedingungen erstellt. Für die Enthakungspfadplanung wird ebenfalls, wie bereits bei der Verhakungserkennung, der Tiefenkartenzuschnitt der verhakten Werkstücksituation verwendet. Dieser kann durch Methoden der Domänenrandomisierung und Domänenanpassung an reale Tiefenbilder angenähert werden. Die Verfahren aus Abschnitt 3.2.1 zur Domänenrandomi-

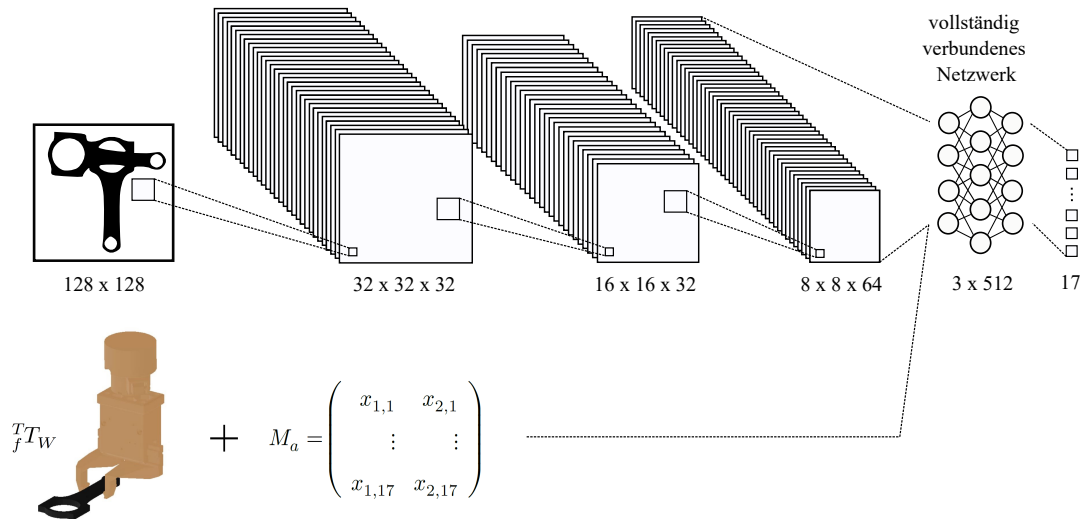


Abbildung 4.17: Netzwerkarchitektur des bestärkenden Lernverfahrens in Anlehnung an Moosmann et al. (2021a)

sierung sowie das Verfahren aus Abschnitt 3.2.3 zur Domänenanpassung werden auch auf die Tiefenkartenzuschnitte für die Enthakungspfadplanung angewendet. Dieser Abschnitt beschreibt, wie die Methoden in den Trainingsprozess der überwachten und bestärkenden Lernverfahren implementiert werden. Die Ergebnisse werden in Kapitel 5 vorgestellt.

#### 4.5.1 Implementierung des Transfers für die überwachten Lernverfahren

Für überwachte Lernverfahren zur Enthakungspfadplanung werden Daten vorab in Simulationsszenen generiert und damit im Nachhinein künstliche, neuronale Netzwerke trainiert. Die Transfermethoden der Domänenrandomisierung und Domänenanpassung können also, unabhängig vom Trainingsprozess angewendet, werden. Nachdem die Datengenerierung in der Simulationsszene abgeschlossen ist, werden die Tiefenkartenzuschnitte durch die Transfermethoden verändert und erst danach einem neuronalen Netzwerk zum Training zur Verfügung gestellt.

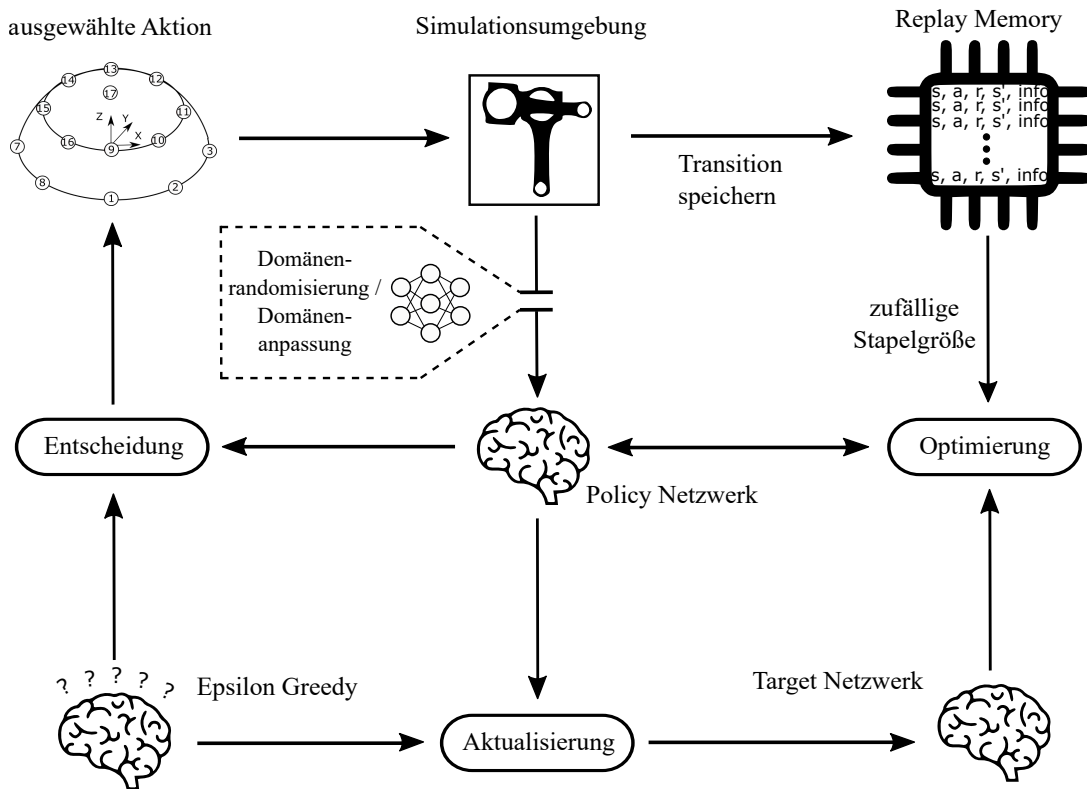


Abbildung 4.18: Trainingsschema des bestärkenden DQN-Agententrainings mit der Implementierung der Transfermethoden in Anlehnung an Moosmann et al. (2021a)

## 4.5.2 Implementierung des Transfers für das bestärkende Lernverfahren

Für das bestärkende Lernverfahren müssen die Transfermethoden in den Trainingsprozess integriert werden. Der DQN-Agent lernt anhand des Tiefenkartenzuschnitts, welcher unmittelbar in der Simulationsszene zum Training des Agenten aufgezeichnet wird. Nach der Aufzeichnung muss deshalb der Tiefenkartenzuschnitt durch die Transfermethoden verändert werden, bevor er dem DQN-Agenten zur Verfügung gestellt wird. Hierfür werden die Domänenrandomisierungsmethoden und die Domänenanpassungsmethode in den Trainingsablauf, wie in Abbildung 4.18 dargestellt, integriert.

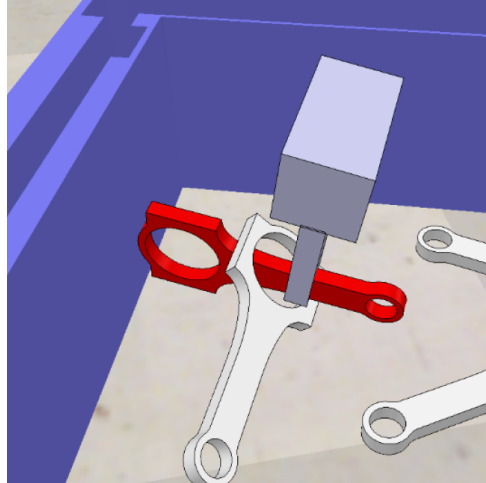


Abbildung 4.19: Beispielhafte Situation von zwei Pleuel, die mit der ausgewählten Greifpose nicht enthakbar sind (Moosmann et al. 2022a).

#### 4.6 Einführung eines Ablehnungskriteriums zur Reduzierung situativ unmöglicher Enthakungsstrategien

In Abhängigkeit der Werkstückgeometrie und der ausgewählten Greifpose am Werkstück kann es dazu kommen, dass mit diesen Gegebenheiten eine Enthakung nicht möglich ist. Abbildung 4.19 zeigt eine solche Situation mit Pleuel. Die Greifpose am Werkstück verhindert, dass das rot gefärbte Pleuel vom weißen getrennt werden kann. Solche Situationen gilt es zu erkennen und zu vermeiden.

Hierfür wird sowohl für die überwachten als auch für die bestärkende Lernmethode ein Verfahren entwickelt. Zugrunde gelegt werden die Bewertungen der Hemisphärenpfadposen der ersten Hemisphäre. Für jede Hemisphärenpfadpose wird durch die jeweilige maschinelle Lernmethode eine Prognose abgegeben. Der Maximalwert  $p_{max}$  der Prognosen für eine verhakte Werkstücksituation in einer Hemisphäre, also der beste aus 17 Werten, wird mit einem Grenzwert  $p_g$  verglichen. Wenn der Maximalwert den Grenzwert nicht überschreitet  $p_{max} \leq p_g$ , liegt mit hoher Wahrscheinlichkeit eine verhakte Werkstücksituation vor, die mit der ausgewählten Greifpose nicht enthakbar ist.

Die Bestimmung des Grenzwertes wird mithilfe der 100 verhakten Werkstücksituationen zur Bestimmung der Konvergenz aus Abschnitt 4.4.1 realisiert. Die 100 Situationen wer-



den in eine Simulationsszene ohne Kiste geladen. Für jede verhakte Situation werden alle definierten Greifposen mit allen Freiheitsgraden verwendet. Die Prognosen der drei maschinellen Lernmethoden zur Enthakungspfadplanung werden für alle drei Hemisphären herangezogen und die jeweils am besten bewertete Pfadpose angefahren. Nach dem optimalen Enthakungspfad wird auf erfolgreiche Enthakung geprüft. Dieser Prozess wird für verschiedene Grenzwerte  $p_g$  im Intervall  $[0,1]$  mit der Schrittweite von  $10^{-3}$  wiederholt. Jeder Pfad wird daraufhin folgendermaßen bewertet:

- *True Positive*: Die Verhakung wurde erfolgreich gelöst und der Maximalwert der Prognosen aus der ersten Hemisphäre überschreitet den Grenzwert.
- *True Negative*: Die Verhakung konnte nicht erfolgreich gelöst werden und der Maximalwert der Prognosen aus der ersten Hemisphäre unterschreitet den Grenzwert.
- *False Positive*: Die Verhakung konnte nicht erfolgreich gelöst werden und der Maximalwert der Prognosen aus der ersten Hemisphäre überschreitet den Grenzwert.
- *False Negative*: Die Verhakung wurde erfolgreich gelöst und der Maximalwert der Prognosen aus der ersten Hemisphäre unterschreitet den Grenzwert

Aus den *True Positive*, den *False Positive* und den *False Negative* Bewertungen wird das  $F_1$ -Maß berechnet. Dieses ist in Abbildung 4.20 für Rundstahlbügel, Pleuel und Haken über dem variierenden Grenzwert aufgetragen. Das Maximum des  $F_1$ -Maßes kann ermittelt werden. Der Grenzwert, der zu diesem Maximum führt, wird als Grenzwert  $p_g$  für die Ablehnung vorgeschlagen.

Wenn eine Verhakung mit der ausgewählten Peripherie, beispielsweise der Greifpose, nicht lösbar ist, werden die Kosten der Greifplanung des verhakten Werkstücks und der Greifpose durch Addition eines Parameters erhöht. Dies hat zur Folge, dass die Heuristik vorrangig andere Werkstücke und Greifposen zum Greifen berücksichtigt. Sollte nur dieses Werkstück mit dieser Greifpose auswählbar und anfahrbar sein, können durch die schlechte Bewertung Notfallstrategien angewendet werden. Beispielsweise wird in einer Notfallstrategie das Werkstück trotzdem angehoben, allerdings wieder über der Kistenmitte fallen gelassen und ein erneuter Greifprozess kann mit der Aufnahme eines Tiefenbildes starten.

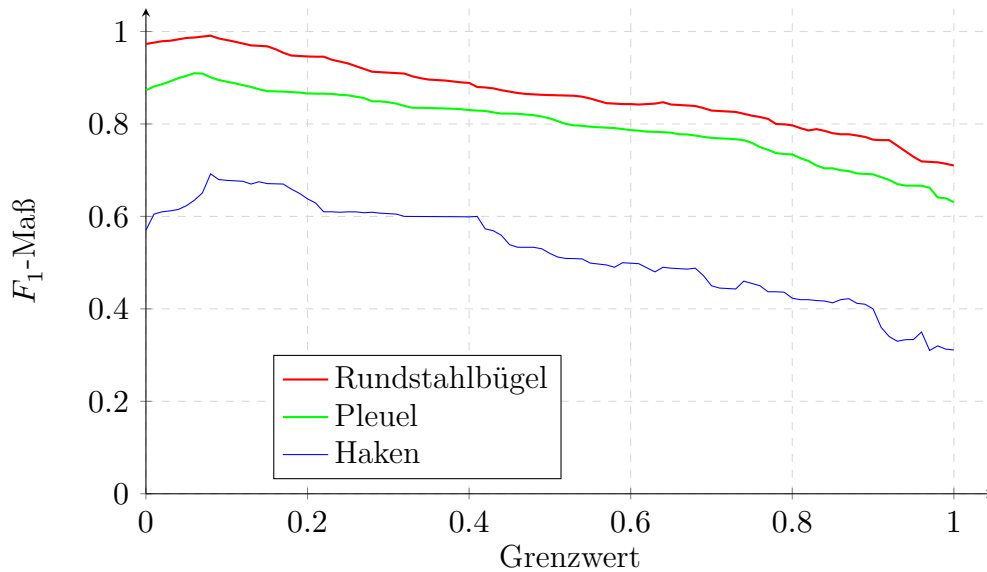


Abbildung 4.20: Bestimmung des Grenzwertes zur Ablehnung situativ unmöglicher Enthakungsstrategien für Rundstahlbügel, Pleuel und Haken anhand des  $F_1$ -Maß.

## 4.7 Einordnung der Enthakungspfadplanung in das Gesamtsystem

Das entstandene Enthakungsverfahren wird als zusätzliches Modul in die Software *BinPicking3d<sup>TM</sup>* implementiert und ist mit der Integration in Abbildung 4.21 dargestellt. Die Integration der Verhakungserkennung in das Gesamtsystem wurde bereits in Abschnitt 3.4 beschrieben. Das Modul zur Enthakungsplanung wird sowohl mit den Werkstücklagen der Objektlageerkennung als auch mit den Tiefenkartenzuschnitten aus der Verhakungserkennung gespeist. Zusätzlich werden von der Greifplanung vordefinierte Greifposen am Werkstück mit Freiheitsgraden bereitgestellt. Weitere Parameter, wie der Hemisphärenradius oder der Grenzwertwert des Ablehnungskriteriums, können vorab bestimmt werden. Nachdem ein verhaktes Werkstück von der Heuristik zur weiteren Betrachtung ausgewählt wurde, wird die Enthakung aktiviert. Die Greifplanung fragt das Enthakungsmodul nach einer möglichen Enthakungspfadpose an. Das Modul schlägt eine Pfadpose auf der ersten Hemisphäre vor. Diese Pfadpose wird an die Greifplanung übergeben, welche mithilfe der Punktwolke auf mögliche Kollisionen mit der Peripherie untersucht. Das Ergebnis dieser Untersuchung wird an das Modul zur Enthakungsplanung übergeben. Sollte die prognostizierte, erste Pfadpose nicht kollisionsfrei anfahrbar sein, so wird von dem Modul zur Enthakungsplanung die nächstbeste Pfadpose vorgeschlagen. Dieser Kommunikations-

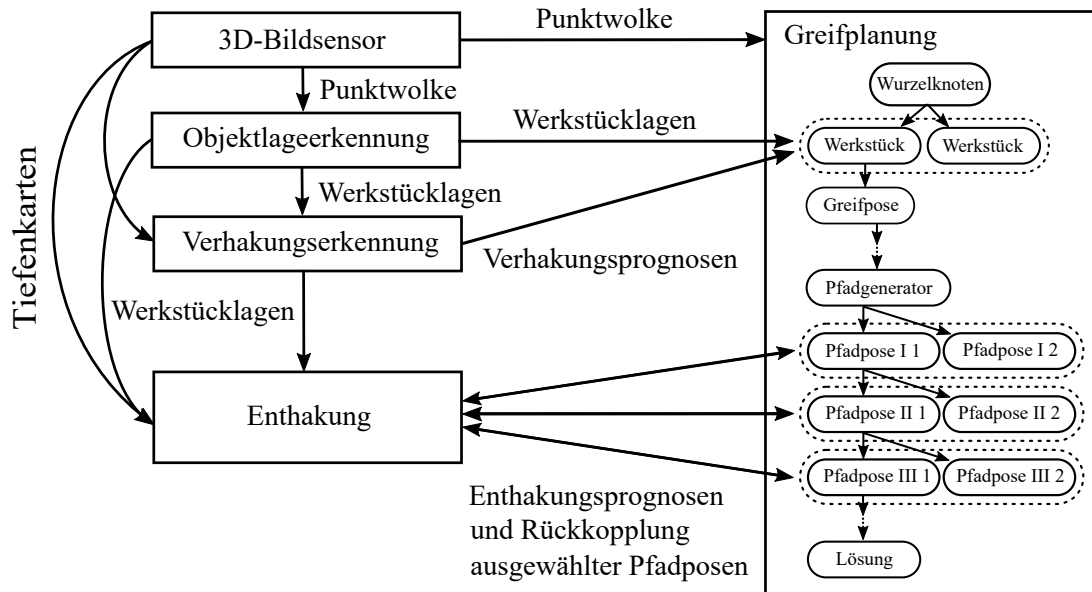


Abbildung 4.21: Ablaufschema und Interaktion einer Enthakungsplanung in einer Griff-in-die-Kiste-Applikation (Moosmann et al. 2021b)

prozess wird solange durchgeführt, bis eine mögliche Pfadpose auf der ersten Hemisphäre gefunden wurde. Nun wiederholt sich der Prozess für die zweite und dritte Hemisphäre, je nachdem wie viele Hemisphären für eine erfolgreiche Enthakung als notwendig prognostiziert wurden. Sollte die Hemisphärenpfadpose mit der Nummer 17 prognostiziert werden, so wird der Enthakungspfad vorzeitig abgebrochen. In diesem Fall wird angenommen, dass bereits eine erfolgreiche Enthakung durch die vorherigen Hemisphären entstanden ist. Die zusätzlich berechneten Pfadposen werden in den Entnahmepfad integriert und an die Steuerung des Industrieroboters übergeben, welcher die Roboterbahn abfährt.

## 4.8 Zusammenfassung

In diesem Kapitel wurde eine Methode zur Enthakungspfadplanung von verhakten Werkstücksituationen in einer Kiste beim Griff-in-die-Kiste entwickelt. Zu Beginn wurde die Ausgangslage und Aufbereitung benötigter Daten beschrieben. Insbesondere wurden hierbei die Daten der Bildverarbeitung und Objektlageerkennung und die Daten der Greifplanung aufgezeigt. Es wurde ein Hemisphärenmodell zur Repräsentation von Bewegungspfaden zur Enthakung vorgestellt. Hierbei wurden Pfadposen auf einer Hemisphäre verteilt und mehrere dieser Hemisphären zur Enthakungspfadplanung erzeugt. Um die Hemisphärenpfadposen auf erfolgreiche Enthakung auszuwählen, wurde eine Annotierung vorgestellt.

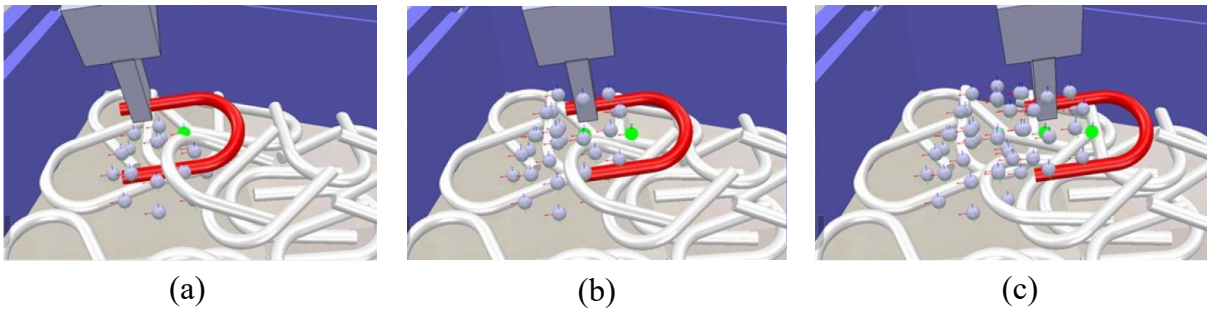


Abbildung 4.22: Ablauf einer Enthakung einer verhakten Werkstücksituation von Rundstahlbügeln in Simulation in der Reihenfolge (a), (b) und (c). (Moosmann et al. 2022b)

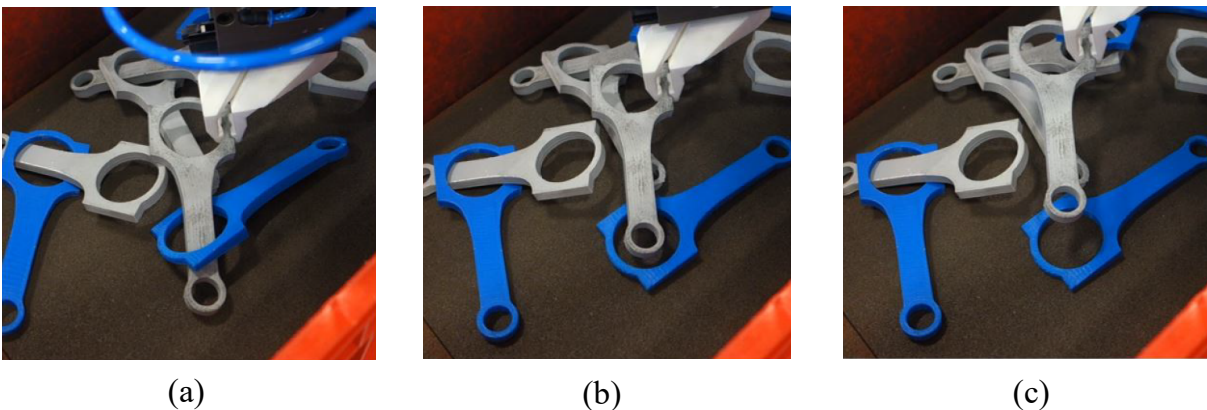


Abbildung 4.23: Ablauf einer Enthakung einer verhakten Werkstücksituation von Pleueln in einer realen Roboterzelle in der Reihenfolge (a), (b) und (c). (Moosmann et al. 2022b)

Die Parameter des Hemisphärenmodells wurden in Versuchsreihen bestimmt und das Modell validiert. Zusätzliche Orientierungsänderungen zu den translatorischen Bewegungen zu den Hemisphärenpfadposen wurden eingeführt. Abbildung 4.22 und Abbildung 4.23 zeigen einen möglichen Enthakungspfad zur Trennung der verhakten Rundstahlbügeln und Pleuel in einer Simulationsumgebung und einer realen Roboterzelle.

Die Enthakungspfadplanung basiert auf maschinellen Lernverfahren. Es wurden zwei überwachte Lernverfahren und ein bestärkendes Lernverfahren vorgestellt. Neben der Datengenerierung für die überwachten Lernverfahren, wurden eine Reihenschaltung von neuronalen Netzwerken und ein zusammengeführtes, neuronales Netzwerk eingeführt und deren Architektur und Training beschrieben. Für die eingeführte, bestärkende Lernmethode wurde das Training eines Agenten dargestellt.

Am Ende des Kapitels wurden der Transfer von der Simulationsumgebung in die Realität für die vorgestellten maschinellen Lernmethoden beschrieben, ein zusätzliches Kriterium zur Ablehnung situativ unmöglicher Enthakungsstrategien eingeführt und die Enthakungspfadplanung in das Griff-in-die-Kiste-Gesamtsystem eingeordnet.



# 5 Ergebnisse und Validierung

In diesem Kapitel werden die in dieser Arbeit entwickelten Verfahren zur Verhakungserkennung und Enthakungspfadplanung in einem Griff-in-die-Kiste-System experimentell validiert. Dazu werden zunächst die Evaluierungsumgebungen in der Simulation und in der Realität vorgestellt. In Abschnitt 5.2 wird das Verfahren der Verhakungserkennung im direkten Transfer von der Simulation in die Realität untersucht. Ferner werden die überwachten Lernverfahren zur Enthakungspfadplanung in Abschnitt 5.3 im direkten Transfer verglichen. Die überwachten Lernverfahren zur Enthakungspfadplanung werden in Abschnitt 5.4 mit dem bestärkenden Lernverfahren im direkten Vergleich validiert. In Abschnitt 5.5 wird der Einfluss von Transfermethoden von der Simulation in die reale Welt auf die Verhakungserkennung und Enthakung evaluiert. Zusätzliche Optimierungen der Enthakungsmethodik werden in Abschnitt 5.6 untersucht und abschließend das Gesamtsystem in Abschnitt 5.7 validiert.

## 5.1 Beschreibung der Evaluierungsumgebung in der Simulation und in der Realität

Zur Evaluierung der entwickelten Verfahren werden sowohl eine Simulationsumgebung als auch eine reale Roboterzelle verwendet. Wie bereits bei der Validierung des Hemisphärenmodells von Bewegungspfaden zur Enthakung in Abschnitt 4.2.4, wird die Simulationsumgebung CoppeliaSim von Coppelia Robotics, Ltd.<sup>1</sup> verwendet. CoppeliaSim ist ein Simulationsprogramm mit Fokus auf die Simulation von Roboterkinematiken. Diese können aus einer Bibliothek geladen und angesteuert werden. Zudem bietet CoppeliaSim die Möglichkeit, Kameras zur Generierung von zweidimensionalen Bildern, wie beispielsweise Tiefenkarten und Punktwolken einzubinden. Durch den Import von STL- oder VRML-Dateien können beliebige Flächenmodelle in Szenen geladen und bewegt, sowie auf Kollisionen geprüft werden. Für die physikalischen Berechnungen werden Physik-Engines in

---

<sup>1</sup><https://www.coppeliarobotics.com/>

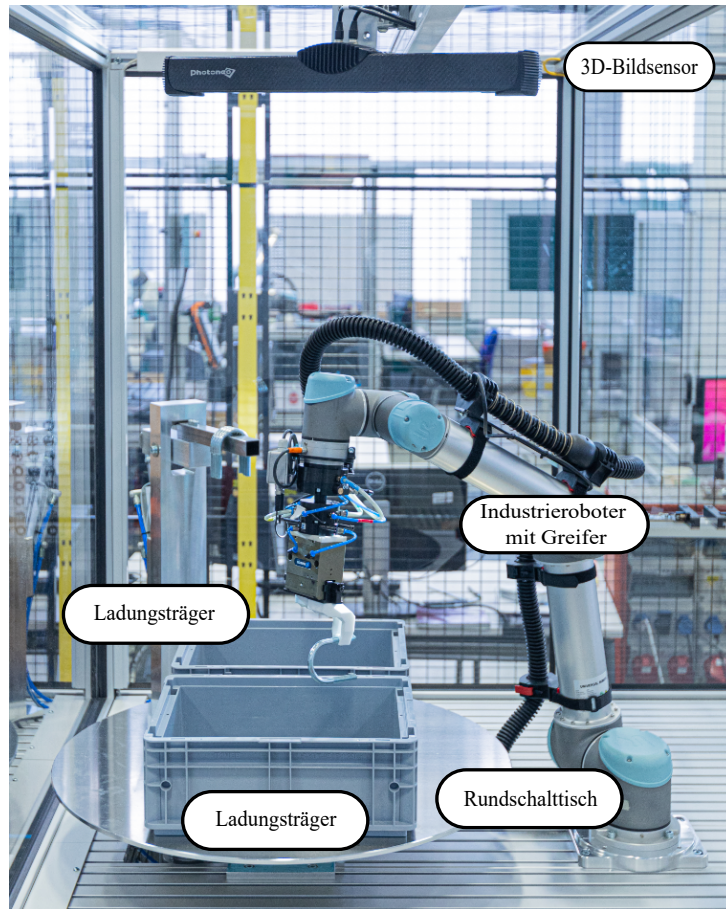


Abbildung 5.1: Versuchsaufbau einer realen Griff-in-die-Kiste-Roboterzelle (Quelle des Originalfotos: Fraunhofer IPA)

der Simulation verwendet. Die Physik-Engine *Bullet 2.83* hat sich empirisch im Vergleich zu den Engines *Newton* und *ODE* als die geeignetste für Griff-in-die-Kiste-Anwendungen herausgestellt. Die Simulation ist nicht deterministisch, was bedeutet, dass bei gleicher Eingabe und Startbedingungen nicht stets dasselbe Ergebnis zu erwarten ist (Claus & Schwill 2001).

In Abbildung 5.1 ist die Roboterzelle für Realversuche abgebildet. In der Roboterzelle befindet sich ein Universal Robot UR10e der Firma Universal Robots A/S<sup>2</sup> mit einem pneumatischen Greifmodul der Firma Schunk GmbH & Co. KG<sup>3</sup>. An das Greifmodul sind zwei Greiferbacken angeschraubt, welche zusammen einen exzentrischen TCP bilden. Die Greiferbacken sind für die Werkstückgeometrien der Rundstahlbügel, Pleuel und Haken optimiert. Zudem befinden sich in der Roboterzelle zwei Ladungsträger, die auf einem

<sup>2</sup><https://www.universal-robots.com/>

<sup>3</sup><https://schunk.com/>



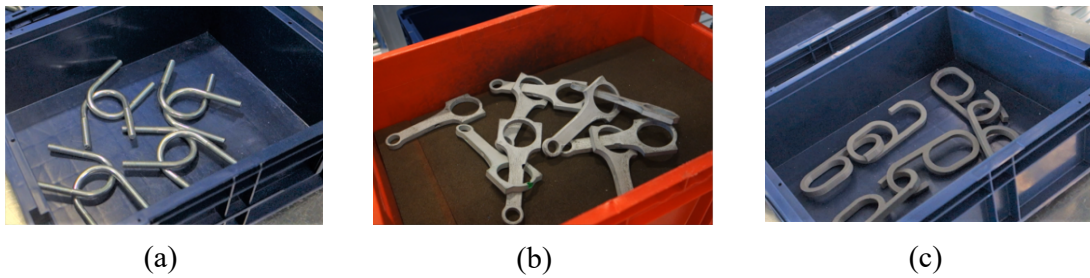


Abbildung 5.2: Verhakte Werkstücksituationen in einer Kiste von (a) Rundstahlbügeln, (b) Pleueln und (c) Haken.

Rundscharltisch abgestellt sind. Aus einem Ladungsträger können Werkstücke entnommen werden und in den zweiten Ladungsträger randomisiert abgeworfen werden. Durch das Drehen des Rundscharltisches kann ein Kreislauf realisiert werden. Über dem Ladungsträger befindet sich ein 3D-Bildsensor der Firma Photoneo s.r.o.<sup>4</sup>. Dieser arbeitet mit einem Streifenprojektionsverfahren, welches Laserstrahlen auf die aufzunehmende Situation projiziert.

Die Software *BinPicking3d<sup>TM</sup>* wird für die Versuche in der realen Roboterzelle verwendet. Nach Eingang der Daten des 3D-Bildsensors wird anhand eines vordefinierten CAD-Modells des Werkstücks ein ansichtenbasiertes Objektlageerkennungsverfahren (Palzkill 2014) eingesetzt. Hierbei werden aus einem Flächenmodell Tiefenbilder aus verschiedenen Ansichten aufgezeichnet und mit Projektionen der Punktwolke in Form von Tiefenkarten verglichen. Durch dieses Verfahren entstehen unbekannte rotatorische und translatorische Ungenauigkeiten in der gefundenen Lage der Werkstücke. Ferner werden nicht alle Werkstücklagen, die theoretisch gefunden werden könnten, lokalisiert. Die Greifplanung der Software *BinPicking3d<sup>TM</sup>* interagiert, wie in Abschnitt 4.7 beschrieben, mit dem entstandenen Verhakungserkennungs- und Enthakungsmodul in den folgenden Versuchsdurchführungen.

In Abbildung 5.2 sind drei Werkstückgeometrien in Kistensituationen dargestellt. Diese drei Werkstückgeometrien dienen als Versuchsgeometrien für die folgenden Versuche. Sie unterscheiden sich in unterschiedlichen verhakbaren Geometriemerkmalen. Beispielsweise weist der Rundstahlbügel eine offene, verhakbare Geometrie auf, während der Pleuel

---

<sup>4</sup><https://www.photoneo.com/>

eine geschlossene, verhakbare Geometrie besitzt. Der Haken stellt eine offene sowie eine geschlossene, verhakbare Geometrie in einem Werkstück dar.

## 5.2 Verhakungserkennung im direkten Transfer

In einem ersten Versuch wird die Methode zur Verhakungserkennung aus Kapitel 3 in einem direkten Transfer untersucht. Hierfür werden Tiefenkartenzuschnitte von verhakten und unverhakten Werkstücksituationen in einer Kiste in einer Simulationsszene aufgezeichnet. Die Anzahl der Werkstücke wird hierfür in Kategorien variiert. Für simulierte Tiefenkartenzuschnitte wird nach der Anzahl 2, 10, 20 und 40 an Werkstücken kategorisiert. Diese Kategorisierung findet für Rundstahlbügel, Pleuel und Haken statt. Zudem werden für jede Werkstückgeometrie reale Tiefenkartenzuschnitte aus einer realen Roboterzelle untersucht. Die Anzahl an Werkstücken in den realen Aufnahmen wird zwischen 2 und 40 randomisiert.

Das CNN, welches für diese Versuchsdurchführung verwendet wird, wurde auf reinen Simulationsdaten trainiert. Das Training ist in Abschnitt 3.3 beschrieben. Zur Auswertung dieses Versuchs werden für jedes Werkstück und für jede Kategorie 1.000 Tiefenkartenzuschnitte verwendet. Die 1.000 Tiefenkartenzuschnitte unterteilen sich in 500 verhakte und 500 unverhakte Beispiele. Das CNN prädiziert für jeden Tiefenkartenzuschnitt eine Verhakungsprognose. Die Ergebnisse werden in einer Konfusionsmatrix ausgewertet. Hierbei wird folgendermaßen unterschieden:

- *True Positive*: Eine Verhakung liegt vor und wurde richtig erkannt.
- *True Negative*: Eine Verhakung liegt nicht vor und wurde richtig erkannt.
- *False Positive*: Eine Verhakung liegt nicht vor und wurde falsch erkannt.
- *False Negative*: Eine Verhakung liegt vor und wurde falsch erkannt.

In Tabelle 5.1 sind die Ergebnisse der Genauigkeit, der Präzision, der Sensitivität und dem  $F_1$ -Maß für alle Werkstücke und Kategorien dargestellt. Die Tabelle zeigt, je weniger Werkstücke in der Kiste vorhanden und damit weniger Werkstücke in dem Tiefenkartenzuschnitt zu sehen sind, desto höher ist die Genauigkeit der Prognose. Auch die Präzision

Tabelle 5.1: Analyse der Methode zur Verhakungserkennung im direkten Transfer für Rundstahlbügel, Pleuel und Haken.

Werkstück	Kategorie (Sim/Real, Anzahl Werkstücke)	Genauigkeit	Präzision	Sensitivität	$F_1$
Rundstahlbügel	Sim, 2	1,00	1,00	1,00	1,00
	Sim, 10	0,98	0,94	0,96	0,95
	Sim, 20	0,94	0,92	0,98	0,95
	Sim, 40	0,88	0,75	0,98	0,85
	Real, random	0,47	0,47	0,80	0,59
Pleuel	Sim, 2	1,00	1,00	1,00	1,00
	Sim, 10	1,00	1,00	1,00	1,00
	Sim, 20	0,95	0,90	0,96	0,93
	Sim, 40	0,86	0,71	0,96	0,82
	Real, random	0,51	0,55	0,80	0,65
Haken	Sim, 2	0,84	0,88	0,92	0,90
	Sim, 10	0,81	0,83	0,89	0,86
	Sim, 20	0,77	0,78	0,82	0,80
	Sim, 40	0,76	0,75	0,81	0,78
	Real, random	0,53	0,53	1,00	0,69

und Sensitivität nehmen mit der zunehmenden Anzahl an Werkstücken ab. Das beste Ergebnis und damit das höchste  $F_1$ -Maß erzielen die simulierten Tiefenkartenzuschnitte mit zwei Werkstücken in der Situation mit 1,00 für die Rundstahlbügel und Pleuel und mit 0,90 für die Haken. Die realen Tiefenkartenzuschnitte schneiden im Vergleich zu den simulierten Tiefenkartenzuschnitten schlechter ab. Eine Begründung hierfür sind fehlende Messdaten aus der realen Anwendung, welche in simulierten Daten nicht vorkommen. Für die Rundstahlbügel wird mit Realdaten eine Genauigkeit von 0,47 und ein  $F_1$ -Maß von 0,59 erzielt. Für die Pleuel unterscheiden sich die Werte mit einer Genauigkeit von 0,51 und einem  $F_1$ -Maß von 0,65 nur minimal von den Werten der Rundstahlbügel. Die Genauigkeit bei Realdaten der Haken beträgt 0,53 und das  $F_1$ -Maß 0,69. Beachtenswert bei den Haken ist der Sensitivitätswert von 1,0 auf realen Datenbeispielen. Der Klassifizierer erkennt sämtliche Situationen als Verhakung und ist in der Lage, alle verhakten Situationen zu finden. Aufgrund der Geometrie der Haken wird ein sonderbarer Schattenwurf beobachtet, welcher durch Transfermethoden auf simulierte Zuschnitte angepasst werden muss.

### 5.3 Vergleich der überwachten Lernverfahren zur Enthakung im direkten Transfer

In diesem Abschnitt erfolgt ein Vergleich der beiden vorgestellten überwachten Lernverfahren zur Enthakung im direkten Transfer in eine reale Roboterzelle. Hierbei findet der Vergleich zwischen der vorgestellten Reihenschaltung von CNNs aus Abschnitt 4.3.2 und dem zusammengeführten CNN aus Abschnitt 4.3.3 statt. Beide Ansätze werden in der beschriebenen, realen Roboterzelle validiert. In eine Kiste werden manuell verhakte Werkstücksituationen gelegt. Hierbei wird die Anzahl an verhakten Situationen in der Kiste zwischen 1 und 10 und die Lage der Verhakungen randomisiert, wobei beide Ansätze die nahezu gleichen Situationen lösen müssen. Die Verhakungserkennung ist dabei ausgeschaltet und jede gefundene Werkstücklage wird als Verhakung angenommen. Insgesamt werden für die Werkstückgeometrien der Rundstahlbügel, Pleuel und Haken jeweils 500 Verhakungen gelegt. Dabei wird das vorgestellte Hemisphärenmodell aus Abschnitt 4.2 mit den aus Abschnitt 4.2.4 beschriebenen Ergebnissen für die Enthakungspfadplanung ohne Orientierungsänderungen verwendet. Die überwachten Lernverfahren wurden, wie in Abschnitt 4.3.4 beschrieben, auf Simulationsdaten trainiert und ohne Domänentransfer in die reale Roboterzelle übertragen. Die Ergebnisse des Versuchs sind in Tabelle 5.2 dargestellt.

Das zusammengeführte CNN schneidet bei den Rundstahlbügeln am besten ab. Mit 239 von 500 gelösten Verhakungen wird eine Enthakungsrate von 47,8% erreicht. Die Reihenschaltung von CNNs löst 235 von 500 verhakten Rundstahlbügeln und erzielt damit eine Enthakungsrate von 47,0%. Von 500 verhakten Pleueln löst die Reihenschaltung der CNNs 208 und damit eine mehr als das zusammengeführte CNN. Die Reihenschaltung erreicht demzufolge eine Enthakungsrate von 47,6%. Eine Enthakung von 125 der 500 verhakten Haken wird durch das zusammengeführte CNN mit einer Enthakungsrate von 25,0% erreicht und ist damit um 0,2% besser als die Reihenschaltung der CNNs.

Tabelle 5.3 zeigt einen Vergleich der beiden überwachten Lernverfahren in der Menge der benötigten Daten und der Dauer der Datengenerierung und des Trainings. Für beide

Tabelle 5.2: Vergleich der angewandten überwachten Lernverfahren (zusammengeführtes CNN und Reihenschaltung von CNNs) zur Enthakung im direkten Transfer in die Roboterzelle für Rundstahlbügel, Pleuel und Haken.

Werkstück	Ansatz	Anzahl Enthakungen	Enthakungsrate
Rundstahlbügel	Zusammenführung	<b>239 / 500</b>	<b>47,8%</b>
	Reihenschaltung	235 / 500	47,0%
Pleuel	Zusammenführung	207 / 500	41,4%
	Reihenschaltung	<b>208 / 500</b>	<b>41,6%</b>
Haken	Zusammenführung	<b>125 / 500</b>	<b>25,0%</b>
	Reihenschaltung	124 / 500	24,8%

Tabelle 5.3: Vergleich der benötigten Daten und der Dauer der Datengenerierung und des Trainings für das zusammengeführte CNN und die Reihenschaltung von CNNs der überwachten Lernverfahren am Beispiel des Pleuels.

Ansatz	Menge benötigter Simulationsdaten	Dauer Datengenerierung	Dauer Training
Zusammenführung	$\approx 20.000$	$\approx 1.500$ Stunden	$\approx 1$ Stunde
Reihenschaltung	$\approx 20.000$	$\approx 1.500$ Stunden	$\approx 1,5$ Stunden

Verfahren sind die Mengen der benötigten Simulationsdaten mit 20.000 Datenbeispielen gleich. Die Dauer der Datengenerierung ist für beide Verfahren mit 1.500 Stunden auf einem Intel Core i9-10980XE mit 4,6GHz und 64GB Arbeitsspeicher ebenfalls identisch. Die Dauer des Trainings des zusammengeführten CNN schneidet mit einer Stunde etwas besser ab als die Reihenschaltung von CNNs, welche für das Training 1,5 Stunden benötigt.

Die Erkenntnisse aus diesem Versuch werden in der nachfolgenden Versuchsdurchführung berücksichtigt. Hierbei wird für jedes Werkstück das jeweils beste überwachte Lernverfahren mit dem bestärkenden Lernverfahren zur Enthakungspfadplanung verglichen.

## 5.4 Vergleich des überwachten und bestärkenden Lernverfahrens zur Enthakung im direkten Transfer

In diesem Abschnitt erfolgt ein Vergleich zwischen den überwachten und bestärkenden Lernverfahren aus Abschnitt 4.4 zur Enthakungspfadplanung in einem direkten Transfer

Tabelle 5.4: Vergleich des aus Abschnitt 5.3 besten überwachten Lernverfahrens (Reihenschaltung von CNNs oder zusammengeführtes CNN) und des bestärkenden Lernverfahrens zur Enthakung im direkten Transfer in die Roboterzelle für Rundstahlbügel, Pleuel und Haken.

Werkstück	Ansatz	Anzahl Enthakungen	Enthakungsrate
Rundstahlbügel	überwacht, Zusammenführung	239 / 500	47,8%
	bestärkend	<b>284 / 500</b>	<b>56,8%</b>
Pleuel	überwacht, Reihenschaltung	208 / 500	41,6%
	bestärkend	<b>210 / 500</b>	<b>42,0%</b>
Haken	überwacht, Zusammenführung	125 / 500	25,0%
	bestärkend	<b>138 / 500</b>	<b>27,6%</b>

in die reale Roboterzelle. Hierfür wird das überwachte Lernverfahren ausgewählt, welches für eine jeweilige Werkstückgeometrie am besten abgeschnitten hat. Der Versuch ist gleich aufgebaut, wie der in Abschnitt 5.3 beschriebene Versuch. Für jedes Werkstück werden 500 Verhakungen manuell mit der Anzahl 1 bis 10 und in der Lage randomisiert in eine Kiste gelegt. Die Verhakungserkennung ist dabei ausgeschaltet und jede gefundene Werkstücklage wird als Verhakung angenommen. Die Enthakungspfadplanung erfolgt nach dem Hemisphärenmodell aus Abschnitt 4.2 ohne Orientierungsänderungen. Die Ergebnisse sind in Tabelle 5.4 dargestellt.

Als bestes überwachtetes Lernverfahren für die Rundstahlbügel wurde das zusammengeführte CNN ermittelt. Das bestärkende Lernverfahren erreicht mit einer Enthakungsrate von 56,8% für die Rundstahlbügel ein deutlich besseres Ergebnis als das überwachte Lernverfahren. Im Vergleich der Reihenschaltung der CNNs und des bestärkenden Lernverfahrens für die Pleuel schneidet das bestärkende Verfahren mit einer Enthakungsrate von 42,0% um 0,4% besser ab als das überwachte Verfahren. Eine Steigerung in der Enthakungsrate von 2,6% erzielt das bestärkende Lernverfahren für die Haken im Vergleich zum überwachten Lernverfahren des zusammengeführten CNN. Die Enthakungsrate des bestärkenden Verfahrens für Haken liegt bei 27,6%. Für alle drei Werkstückgeometrien liefert das bestärkende Lernverfahren damit bessere Ergebnisse als die ausgewählten, überwachten Lernverfahren.

Tabelle 5.5: Vergleich der benötigten Daten, der Dauer der Datengenerierung und des Trainings sowie der Gesamtdauer für die überwachten und für das bestärkende Lernverfahren am Beispiel des Pleuels.

Ansatz	Menge benötigter Simulationsdaten	Dauer Datengenerierung	Dauer Training	Gesamtdauer
überwacht	$\approx 20.000$	$\approx 1.500$ Stunden	$\approx 1-1,5$ Stunden	$\approx 1.500$ Stunden
bestärkend	$\approx 7.500$	0 Stunden	$\approx 175$ Stunden	$\approx 175$ Stunden

Ein Grund hierfür ist die stärker ausgeprägte Exploration des Aktionsraums des Agenten für das bestärkende Lernen im Vergleich zur Datengenerierung für das überwachte Lernen. Bei der Datengenerierung für die überwachten Lernverfahren werden jeweils nur die besten Hemisphärenpfadposen angefahren und von diesen weitere Hemisphären erstellt. Somit ist nicht jede Kombination eines Enthakungspfades im Datensatz enthalten. Die Reduzierung in der Datengenerierung wurde vorgenommen, um schneller mögliche Enthakungspfade für eine verhakte Werkstücksituation zu finden. Durch die Reduzierung werden jeweils nur  $17 + 17 + 17 = 51$  Pfadposen evaluiert, während bei einer vollständigen Exploration  $17^3 = 4913$  Pfadposen hätten evaluiert werden müssen, was zu einer längeren Dauer der Datengenerierung geführt hätte. Durch die angewandte *Epsilon Greedy Strategie* erfährt der Agent somit einen größeren Aktionsraum und mehr Pfadmöglichkeiten. Um den Aufwand der verglichenen Lernverfahren in der Datengenerierung und Trainingszeit gegenüberzustellen, sind in Tabelle 5.5 für das Pleuel Unterschiede dargestellt.

Wie bereits in Tabelle 5.3 festgestellt, benötigen die überwachten Lernverfahren 20.000 generierte Daten, um im Training zu konvergieren. Das bestärkende Lernverfahren benötigt hingegen keine Trainingsdaten, welche im Vorhinein generiert werden müssen. Somit benötigt das bestärkende Lernverfahren auch keine Zeit für eine Datengenerierung vorab des Trainings, wohingegen die überwachten Lernverfahren 1.500 Stunden benötigen. Die Dauer des Trainings des bestärkenden Lernverfahrens ist hingegen mit 175 Stunden deutlich länger als das Training der überwachten Lernverfahren mit einer Dauer von 1 bis 1,5 Stunden. Somit erreicht das überwachte Lernverfahren eine Konvergenz des Trainings in einer Gesamtdauer von 1.500 Stunden, während das bestärkende Lernverfahren hierfür nur

175 Stunden benötigt. Durch die festgestellte, höhere Enthakungsrate und die deutlich geringere Gesamtdauer, die für eine Konvergenz im Training für die Enthakungspfadplanung des bestärkenden Lernverfahrens notwendig ist, wird dieses Verfahren für alle folgenden Versuche verwendet.

## 5.5 Einfluss von Transfermethoden auf die Verhakungserkennung und Enthakung

In diesem Abschnitt wird der Einfluss von den angewandten Transfermethoden von der Simulation in die reale Roboterzelle im Vergleich zu einem direkten Transfer für die Methoden der Verhakungserkennung und Enthakung untersucht. Neben dem Einfluss verschiedener Randomisierungsparameter werden die Optimierungen der Methoden der Verhakungserkennung und Enthakung mit zusätzlicher Domänenrandomisierung und Domänenanpassung evaluiert.

### 5.5.1 Einfluss verschiedener Randomisierungsmethoden

In Abschnitt 3.2.1 wurden verschiedene Methoden zur Domänenrandomisierung vorgestellt. Die Domänenrandomisierung ermöglicht die in der Simulation perfekt generierten Tiefenkartenzuschnitte so zu verändern, dass sie realen Tiefenkartenzuschnitten ähnlich sind. In folgendem Versuch werden die Methoden des translatorischen und rotatorischen Versatzes, die Helligkeitsrandomisierung und das Gauß'sche Rauschen angewandt und deren Einfluss auf die Methode der Verhakungserkennung evaluiert. Zudem wird die Eignung der Methode des Inpaintings aus Abschnitt 3.2.2 untersucht. Die simulierten Trainingsdaten des neuronalen Netzes werden vor dem Training durch die jeweilige Randomisierungsmethode verändert. Hierbei werden für den Translationsversatz  $\sigma_T = 5$ , für den Rotationsversatz  $\sigma_R = 10$ , für das Gauß'sche Rauschen  $\sigma_G = 0,01$  und für die Helligkeitsrandomisierung  $\sigma_H = 50$  verwendet. Der Versuch wird exemplarisch mit der Werkstückgeometrie des Pleuels durchgeführt. Zur Evaluierung werden 500 Tiefenkartenzuschnitte von verhakten und 500 Tiefenkartenzuschnitte von nicht verhakten Werkstücksituationen der Pleuel in einer Kiste von einem realen 3D-Bildsensor verwendet. Nach dem Training werden die realen Tiefenkartenzuschnitte dem jeweils trainierten, neuronalen Netzwerk zur Prognose



Tabelle 5.6: Untersuchung der Eignung verschiedener Randomisierungsmethoden für die Verhakungserkennung am Beispiel des Pleuels.

Methode	Genauigkeit	Präzision	Sensitivität	$F_1$	$\Delta F_1$
Direkter Transfer	0,51	0,55	0,80	0,65	0,00
+ Inpaiting	0,46	0,37	0,88	0,52	- 0,13
+ Translation	0,40	0,43	0,77	0,55	- 0,10
+ Gauß'sches Rauschen	0,73	0,59	0,88	0,71	+ 0,06
+ Rotation	0,65	0,59	0,91	0,72	+ 0,07
+ Helligkeit	0,70	0,64	0,86	0,74	+ 0,09
+ Komb. alle	0,75	0,69	0,89	0,78	+ 0,13
<b>+ Komb. G+R+H</b>	<b>0,83</b>	<b>0,77</b>	<b>0,95</b>	<b>0,85</b>	<b>+ 0,20</b>

übergeben. Neben den alleinigen Randomisierungsmethoden werden auch Kombinationen dieser untersucht. Die Auswertung erfolgt in einer Konfusionsmatrix wie in Abschnitt 5.2. Die Ergebnisse sind in Tabelle 5.6 dargestellt.

Die Verwendung der Randomisierungsparameter wird mit einem direkten Transfer aus Abschnitt 5.2, ohne Anwendung jeglicher Transfermethodik, verglichen. Die Methode des Inpaitings und der translatorische Versatz wirken sich jeweils negativ auf die Verhakungserkennung aus. Inpaiting verursacht eine Reduzierung der Genauigkeit um 0,05 zu 0,46. Zudem wird das  $F_1$ -Maß um 0,13 reduziert. Der translatorische Versatz verursacht eine Reduzierung zum direkten Transfer um 0,11 zu 0,40. Im  $F_1$ -Maß wirkt sich dies auf eine Reduzierung um -0,10 aus. Einen positiven Einfluss hingegen erzielen die Methoden des Gauß'schen Rauschens, dem rotatorischen Versatz und der Helligkeitsrandomisierung. Das Gauß'sche Rauschen erhöht das  $F_1$ -Maß um 0,06, der rotatorische Versatz um 0,07 und die Helligkeitsrandomisierung um 0,09. Die größte Genauigkeit wird hierbei durch das Gauß'sche Rauschen mit 0,73 erzielt. Eine Kombination aus allen vorgestellten Randomisierungsmethoden resultiert in einer Genauigkeit von 0,75 und einer Erhöhung des  $F_1$ -Maß um 0,13. Die beste Kombination besteht aus Gauß'schem Rauschen, dem rotatorischen Versatz und der Helligkeitsrandomisierung. Für diese Kombination wird eine Genauigkeit von 0,83, ein  $F_1$ -Maß von 0,85 und damit eine Erhöhung des  $F_1$ -Maß um 0,20 im Vergleich zum

Tabelle 5.7: Vergleich des direkten Transfers mit der Domänenrandomisierung und Domänenanpassung für die Verhakungserkennung von Rundstahlbügel, Pleuel und Haken.

Werkstück	Ansatz	Genauigkeit	Präzision	Sensitivität	$F_1$	$\Delta F_1$
Rundstahlbügel	Direkter Transfer	0,47	0,47	0,80	0,59	
	Randomisierung	0,78	<b>0,80</b>	0,83	0,82	+0,23
	Anpassung	<b>0,79</b>	0,73	<b>0,98</b>	<b>0,83</b>	<b>+0,24</b>
Pleuel	Direkter Transfer	0,51	0,55	0,8	0,65	
	Randomisierung	<b>0,83</b>	<b>0,77</b>	<b>0,95</b>	<b>0,85</b>	<b>+0,20</b>
	Anpassung	0,82	0,76	0,93	0,84	+0,19
Haken	Direkter Transfer	0,53	0,53	<b>1,00</b>	0,69	
	Randomisierung	0,67	0,59	0,79	0,68	-0,01
	Anpassung	<b>0,79</b>	<b>0,75</b>	0,90	<b>0,82</b>	<b>+0,13</b>

direkten Transfer erreicht. Die vorgestellte Kombination aus Gauß'schem Rauschen, dem rotatorischen Versatz und der Helligkeitsrandomisierung wird in allen weiteren Versuchen verwendet.

### 5.5.2 Optimierung der Methode zur Verhakungserkennung

Die in Abschnitt 5.5.1 ermittelten, besten Randomisierungsmethoden für die Verhakungserkennung werden nun in einem Vergleich mit dem direkten Transfer und der in Abschnitt 3.2.3 vorgestellten Domänenanpassungsmethode verglichen. Hierbei werden alle drei Werkstückgeometrien der Rundstahlbügel, der Pleuel und der Haken betrachtet. Für jedes Werkstück und jede Transfermethode werden jeweils 500 Tiefenkartenzuschnitte mit einer verhakten und 500 Tiefenkartenzuschnitte mit einer unverhakten Werkstücksituation aus einer realen Roboterzelle dem neuronalen Netzwerk zur Prognose übergeben. Zuvor wurde das neuronale Netzwerk mit den durch die Transfermethode veränderten Daten trainiert. Die Auswertung erfolgt mit der Konfusionsmatrix, wie in Abschnitt 5.2 beschrieben. Die Ergebnisse sind in Tabelle 5.7 dargestellt.

Die Ergebnisse zeigen, dass bei den Rundstahlbügeln und den Haken die Domänenanpassung die beste Transfermethode darstellt. Bei den Rundstahlbügeln wird eine Verbesserung des  $F_1$ -Maß um 0,24 im Vergleich zum direkten Transfer erreicht. Bei den Haken beträgt

die Verbesserung des  $F_1$ -Maßes 0,13. Somit kommt die Domänenanpassung bei den Rundstahlbügeln auf ein  $F_1$ -Maß von 0,83 und bei den Haken auf ein  $F_1$ -Maß von 0,82. Für die Pleuel stellt sich die Domänenrandomisierung als am besten geeignet heraus. Im Vergleich zum direkten Transfer wird das  $F_1$ -Maß um 0,20 auf 0,85 erhöht. Bei der Versuchsdurchführung konnte beobachtet werden, dass die Objektlagererkennung bei den Pleueln oftmals Werkstücklagen mit kleinen, rotatorischen Versätzen um die  $z$ -Achse des  $K_W$  lokalisiert. Diese Versätze können von dem verwendeten CycleGAN nur geringfügig ausgeglichen werden, da das CycleGAN auf Basis perfekt simulierter Tiefenkarten verrauscht. Die Versätze sind in den domänenrandomisierten Tiefenkartenzuschnitten deutlicher ausgeprägt. Dieses Verhalten stellt einen Grund dar, warum die Domänenrandomisierung die bessere Transfermethode für die Pleuel darstellt. In den folgenden Versuchen wird die Domänenanpassung für die Verhakungserkennungsmethode der Rundstahlbügel und Haken und die Domänenrandomisierung für die Pleuel verwendet.

### 5.5.3 Optimierung der Methode zur Enthakungspfadplanung

Im folgenden Versuch wird der direkte Transfer der Enthakungspfadplanung mit der Domänenrandomisierung und Domänenanpassung für die drei Werkstückgeometrien der Rundstahlbügel, der Pleuel und der Haken verglichen. Hierfür wird das bestärkende Lernverfahren zur Enthakungspfadplanung in Abhängigkeit der Transfermethode, wie in Abschnitt 4.5.2 beschrieben, trainiert. Anschließend werden in einer realen Roboterzelle 500 verhakte Werkstücksituationen in randomisierter Lage und Anzahl an Verhakungen von 1 bis 10 manuell in eine Kiste gelegt und durch den Roboter manipuliert. Die Verhakungserkennung wird für diesen Versuch nicht aktiviert und jedes lokalisierte Werkstück wird als verhakt angenommen. Die Ergebnisse werden in Tabelle 5.8 vorgestellt.

Die Ergebnisse zeigen, wie bereits bei der Verhakungserkennung in Abschnitt 5.5.2, dass sich die Transfermethode der Domänenanpassung für die Rundstahlbügel und für die Haken am besten eignet. Von 500 möglichen verhakten Werkstücksituationen der Rundstahlbügel wurden 424 erfolgreich enthakt und damit eine Enthakungsrate von 84,8% erzielt. Die Verbesserung zum direkten Transfer beträgt 28,0%. Die Domänenrandomisierung erreicht nur eine Enthakungsrate von 65,6% und damit eine Steigerung um 8,8%. Für die Haken

Tabelle 5.8: Vergleich des direkten Transfers mit der Domänenrandomisierung und Domänenanpassung für die Enthakungspfadplanung von Rundstahlbügeln, Pleueln und Haken.

Werkstück	Transfermethode	Anzahl Enthakungen	Enthakungsrate	Verbesserung zum direkten Transfer
Rundstahlbügel	Direkter Transfer	284 / 500	56,8%	
	Domänenrandomisierung	328 / 500	65,6%	8,8%P
	Domänenanpassung	<b>424 / 500</b>	<b>84,8%</b>	<b>28,0%P</b>
Pleuel	Direkter Transfer	210 / 500	42,0%	
	Domänenrandomisierung	<b>396 / 500</b>	<b>79,2%</b>	<b>37,2%P</b>
	Domänenanpassung	334 / 500	66,8%	24,8%P
Haken	Direkter Transfer	138 / 500	27,6%	
	Domänenrandomisierung	142 / 500	28,4%	0,8%P
	Domänenanpassung	<b>178 / 500</b>	<b>35,6%</b>	<b>8,0%P</b>

erreicht die Domänenanpassung eine Steigerung von 8,0% und eine Enthakungsrate von 35,6%, während die Domänenrandomisierung nur eine Steigerung von 0,8% erzielt. Bei den Pleueln erreicht die Domänenrandomisierung mit einer Enthakungsrate von 79,2% und einer Steigerung zum direkten Transfer von 37,2% das beste Ergebnis. In den folgenden Versuchen werden für die Rundstahlbügel und die Haken die Domänenanpassung und für die Pleuel die Domänenrandomisierung verwendet.

## 5.6 Validierung zusätzlicher Optimierungen der Enthakungsmethodik

In diesem Abschnitt werden zusätzliche Optimierungen der Enthakungsmethodik untersucht. Neben dem Einfluss zusätzlicher Rotationen an den möglichen Hemisphärenpfadpo- sen auf den Enthakungserfolg wird der Einfluss eines Ablehnungskriteriums für unmögliche Enthakungsperipherie evaluiert.

Tabelle 5.9: Validierung des Einflusses zusätzlicher Orientierungsänderungen an den Pfadposen der Hemisphären anhand des Vergleichs der Rotationsmodelle (a) und (b) mit der reinen Translation. Auswertung aller möglicher Enthakungspfade für Rundstahlbügel, Pleuel und Haken in Simulation.

Werkstück	Translation	Rotationsmodell (a)	Rotationsmodell (b)
Rundstahlbügel	77.134 / 491.300 15,7%	102.190 / 491.300 20,8%	<b>108.577</b> / 491.300 <b>22,1%</b>
Pleuel	49.621 / 491.300 10,1%	126.755 / 491.300 25,8%	<b>180.307</b> / 491.300 <b>36,7%</b>
Haken	24.565 / 491.300 5,0%	<b>56.008</b> / 491.300 <b>11,4%</b>	43.726 / 491.300 8,9%

### 5.6.1 Einfluss zusätzlicher Orientierungsänderungen an den Pfadposen der Hemisphären

In Abschnitt 4.2.5 wurden zusätzliche Rotationen der gegriffenen Werkstücke an den jeweiligen Hemisphärenpfadposen eingeführt. Um deren Einfluss auf die Enthakungsrate für die drei Werkstücke zu ermitteln, wurde zuerst ein Versuch in der Simulationsumgebung durchgeführt. In diesem Versuch wird für jede Werkstückgeometrie das geeignetste Rotationsmodell ausgewählt und dieses im Vergleich zur reinen Translation zu den Hemisphärenpfadposen betrachtet. Hierfür wurden die 100 verhakten Werkstücksituationen des Konvergenzkriteriums des bestärkenden Lernverfahrens verwendet (vgl. Abschnitt 4.4.1). Diese wurden nacheinander in die Simulationsszene geladen und alle Pfadmöglichkeiten abgefahren. Nach einer vollständigen Bewegung wurde auf Enthakung geprüft. Somit entstehen bei 17 Hemisphärenpfadposen in einer Hemisphäre und insgesamt 3 Hemisphären für eine verhakte Werkstücksituation 4.913 Pfadmöglichkeiten. Diese Pfadmöglichkeiten multipliziert mit der Anzahl 100 an verhakten Werkstücksituationen ergibt eine Anzahl an möglichen Pfaden von insgesamt 491.300. Die Ergebnisse des Versuches sind in Tabelle 5.9 abgebildet.

Für die Rundstahlbügel liefert das Rotationsmodell (b) die besten Ergebnisse. 108.577 von 491.300 Pfadmöglichkeiten führen zu einer erfolgreichen Enthakung, was 22,1% entspricht. Damit sind mit Rotationsmodell (b) 1,3% mehr Pfade möglich als mit Rotationsmodell

Tabelle 5.10: Enthakungsversuche in der realen Roboterzelle mit zusätzlichen Orientierungsänderungen an den Hemisphärenpfadposen für Rundstahlbügel, Pleuel und Haken mit den Rotationsmodellen (a) und (b) im Vergleich zu einer reinen translatorischen Bewegung.

Werkstück	Rotationsmodell	Anzahl Enthakungen	Enthakungsrate / %	Verbesserung zu Translation / Prozentpunkte
Rundstahlbügel	nur Translation	424 / 500	84,8	
	Rotationsmodell (b)	490 / 500	<b>98,0</b>	<b>13,2</b>
Pleuel	nur Translation	396 / 500	79,2	
	Rotationsmodell (b)	440 / 500	<b>88,0</b>	<b>8,8</b>
Haken	nur Translation	178 / 500	35,6	
	Rotationsmodell (a)	215 / 500	<b>43,0</b>	<b>7,4</b>

(a). Im Vergleich zu einer reinen translatorischen Bewegung zu den Hemisphärenpfadposen steigert Rotationsmodell (b) die Enthakungsmöglichkeit um 6,4 Prozentpunkte. Mit 180.307 und damit 36,7% möglicher Enthakungspfade schneidet Rotationsmodell (b) auch bei den Pleueln am besten ab. Im Vergleich zur reinen Translation steigert das Rotationsmodell (b) den möglichen Enthakungspfadanteil um 26,6 Prozentpunkte. Für die Haken erzielt Rotationsmodell (a) mit 56.008 von 491.300 möglichen Pfaden das beste Ergebnis. Mit 11,4% steigert das Rotationsmodell (a) die Enthakungsmöglichkeiten um 6,4 Prozentpunkte zur reinen Translation.

In einem weiteren Versuch werden die für die Werkstückgeometrien ausgewählten Rotationsmodelle in der realen Roboterzelle verwendet. Hierfür werden jeweils 500 verhakte Werkstücksituationen in randomisierter Lage und Anzahl von 1 bis 10 in eine Kiste gelegt. Die Verhakungserkennung ist deaktiviert und jede gefundene Werkstücklage wird als verhakt angenommen. Die prognostizierten Hemisphärenpfadposen mit den zusätzlichen Orientierungsänderungen werden nach einem erfolgreichen Griff von dem eingesetzten Roboter abgefahren. Die Ergebnisse des Versuchs sind in Tabelle 5.10 aufgeführt.

Für die Rundstahlbügel konnte die Enthakungsrate durch das zusätzliche Rotationsmodell von 84,8% um 13,2 Prozentpunkte auf insgesamt 98,0% im Vergleich zur reinen Translation

gesteigert werden. Eine Steigerung von 8,8 Prozentpunkte auf eine Enthakungsrate von 88,0% wurde bei den Pleueln durch das Rotationsmodell (b) erzielt. Die Enthakungsrate der Haken wird durch Rotationsmodell (a) um 7,4 Prozentpunkte auf 43,0% gesteigert. In der Validierung des Gesamtsystems werden die zusätzlichen Orientierungsänderungen an den Hemisphärenpfadposen durch die ausgewählten Rotationsmodelle verwendet und berücksichtigt.

### 5.6.2 Einfluss eines Ablehnungskriteriums

In Abschnitt 4.6 wurde ein zusätzliches Ablehnungskriterium zur Reduzierung von unmöglicher Enthakungsperipherie eingeführt. Das Ablehnungskriterium soll beispielsweise dazu führen, Greifposen am Werkstück, die eine Enthakung unmöglich machen, zu vermeiden. Um den Einfluss eines zusätzlichen Ablehnungskriteriums auf die Enthakungsrate zu ermitteln, wird ein Versuch in der realen Roboterzelle für die Rundstahlbügel, Pleuel und Haken durchgeführt. Der ermittelte Ablehnungsgrenzwert  $p_g$  beträgt für die Rundstahlbügel  $p_g = 0,063$ , für die Pleuel  $p_g = 0,041$  und für die Haken  $p_g = 0,059$ . Für jede Werkstückgeometrie werden 500 verhakte Werkstücksituationen in randomisierter Lage und Anzahl von 1 bis 10 manuell in eine Kiste gelegt. Die Verhakungserkennung wird wieder deaktiviert und jedes Werkstück wird als verhakt angenommen. Wenn das Grenzwertkriterium  $p_{max} \leq p_g$  nicht erfüllt ist, fährt der Roboter die prognostizierten Hemisphärenpfadposen ab. Wenn es erfüllt wird, liegt vermutlich eine unlösbare, verhakte Werkstücksituation vor und die Kosten des Werkstückes werden in der heuristischen Greifplanung erhöht, sodass ein anderes Werkstück oder eine andere Greifpose ausgewählt werden kann. Die Ergebnisse sind in Tabelle 5.11 dargestellt.

Die Enthakungsrate der Rundstahlbügel wird durch die zusätzliche Ablehnung durch den Grenzwert nur geringfügig um 0,2 Prozentpunkte auf 98,2% gesteigert. Bei den Pleueln hingegen wird eine Steigerung von 5,2 Prozentpunkte auf eine Enthakungsrate von 93,4% erzielt. Mit einer Enthakungsrate von 44,6% mit der Grenzwertbetrachtung für Haken wird eine Steigerung von 1,4 Prozentpunkte erreicht.

Tabelle 5.11: Validierung des Einflusses eines zusätzlichen Ablehnungskriteriums zur Reduzierung unmöglicher Enthakungsperipherie in der realen Roboterzelle für Rundstahlbügel, Pleuel und Haken.

Werkstück	Grenzwert $p_g$	Anzahl Enthakungen	Enthakungsrate / %	Verbesserung / Prozentpunkte
Rundstahl- bügel	0,0	490 / 500	98,0	
	0,063	491 / 500	<b>98,2</b>	<b>0,2</b>
Pleuel	0,0	441 / 500	88,2	
	0,041	467 / 500	<b>93,4</b>	<b>5,2</b>
Haken	0,0	216 / 500	43,2	
	0,059	223 / 500	<b>44,6</b>	<b>1,4</b>

Tabelle 5.12: Ergebnistabelle des ungepaarten t-Tests zur Überprüfung der statistischen Signifikanz der Ablehnungskriterien zur Reduzierung von unmöglicher Enthakungsperipherie

Werkstück	Grenzwert $p_g$	Mittelwert $\bar{x}$	Standardabweichung $S$	t-Wert	p-Wert ( $\alpha = 0,05$ )	p-Wert ( $\alpha = 0,01$ )
Rundstahl- bügel	0,0	0,98	$\approx 0,0158$	$\approx 0,1277$	$\approx 0,8985$	$\approx 0,9453$
	0,063	0,982	$\approx 0,0157$			
Pleuel	0,0	0,882	$\approx 0,024$	$\approx 2,688$	$\approx 0,0076$	$\approx 0,0038$
	0,041	0,934	$\approx 0,0192$			
Haken	0,0	0,432	$\approx 0,0247$	$\approx 0,5592$	$\approx 0,5761$	$\approx 0,5788$
	0,059	0,446	$\approx 0,025$			

Um die statistische Signifikanz der Ergebnisse aus Tabelle 5.11 zu ermitteln, wird ein ungepaarter t-Test mit zwei unabhängigen Stichproben durchgeführt. Die Ergebnisse des Signifikanztests sind in Tabelle 5.12 dargestellt. Als Nullhypothese wird formuliert, dass die Berücksichtigung des Grenzwerts keinen Einfluss auf die Enthakungsrate hat. Die vorgestellten Werte aus Tabelle 5.11 werden zur Ermittlung der Mittelwerte  $\bar{x}$  und der Standardabweichungen  $S$  verwendet. Für jede Werkstückgeometrie werden zwei Signifikanzniveaus  $\alpha = 0,05$  und  $\alpha = 0,01$  untersucht.

Der t-Test zeigt, dass die p-Werte für die Rundstahlbügel und für die Haken das Signifikanzniveau überschreiten und damit für beide Werkstückgeometrien die Nullhypothese



bestätigt ist. Für die Pleuel kann hingegen die Nullhypothese abgelehnt und eine statistische Signifikanz des Ablehnungsgrenzwerts auf die Enthakungsrate festgestellt werden. Die Ergebnisse zeigen, dass bestimmte Werkstückgeometrien, wie beispielsweise die Pleuel, anfälliger für unmögliche Enthakungsperipherien sind und deshalb die Einführung eines Ablehnungskriteriums durchaus sinnvoll ist. Die Grenzwertbetrachtung wird in der Validierung des Gesamtsystems berücksichtigt.

## 5.7 Validierung des Gesamtsystems

Zum Abschluss wird nun mit allen bisherigen Ergebnissen das Gesamtsystem in der realen Roboterzelle validiert. Hierfür werden die Ergebnisse der Verhakungserkennung aus Abschnitt 5.2, der Validierung des Hemisphärenmodells aus Abschnitt 4.2.4, des Vergleichs der überwachten und bestärkenden Lernverfahren für die Enthakungspfadplanung aus Abschnitt 5.4 und der Einfluss von Transfermethoden auf die Verhakungserkennung und Enthakung aus Abschnitt 5.5 berücksichtigt. Die Validierungsergebnisse zusätzlicher Optimierungen der Enthakungsmethodik aus Abschnitt 5.6 werden ebenfalls betrachtet.

Für diesen Versuch wurde eine Kiste mit 30 Werkstücken befüllt und auf den Rundschalttisch neben eine leere Kiste platziert. Das Roboterprogramm wurde so konzipiert, dass ein Endloskreislauf möglich ist. Der Roboter sendet eine Anfrage an die Griff-in-die-Kiste-Software, welche eine Aufnahme des 3D-Bildsensors auslöst. Die Software verarbeitet die Tiefeninformationen und übergibt die Anfahrts- und Entnahmepfadposen an die Robotersteuerung. Der Roboter greift das Werkstück und fährt die Posen ab. Anschließend wirft er es in einer randomisierten Position über der leeren Kiste ab. Dieser Prozess wird so lange wiederholt, bis die Kiste vollständig geleert wurde oder wenn nach drei erfolglosen Versuchen, keine Objektlage oder Greifpose gefunden wurde. Der Rundschalttisch wird im Anschluss um 180° gedreht und der Prozess kann mit der gefüllten Kiste erneut starten.

Die Griff-in-die-Kiste-Software, die in diesem Versuch verwendet wird, ist *BinPicking3d<sup>TM</sup>* und wird in kurzer Form als *bp3<sup>TM</sup>* bezeichnet. Die Software ohne Verwendung eines Moduls zur Verhakungserkennung und Enthakungspfadplanung wird als Standard *bp3<sup>TM</sup>*

Tabelle 5.13: Validierung des Griff-in-die-Kiste-Gesamtsystems ohne Verhakungserkennung und Enthakung als Standard  $bp3^{TM}$  und im Domänen- und direkten Transfer mit Verhakungserkennung und Enthakung für Rundstahlbügel, Pleuel und Haken in der realen Roboterzelle.

Werkstück	Ansatz	Anzahl Greifversuche	Anzahl Verhakungen	Verhakungen erkannt (%)	Enthakungen
Rundstahlbügel	Standard $bp3^{TM}$	6.311	1.566	0 (0,0%)	170
	Direkter Transfer	5.429	1.332	793 (59,54%)	452
	Domänentransfer	6.299	1.589	<b>1.249 (78,6%)</b>	<b>1.224</b>
Pleuel	Standard $bp3^{TM}$	5.985	1.058	0 (0,0%)	102
	Direkter Transfer	6.034	1.212	756 (62,38%)	509
	Domänentransfer	6.089	1.283	<b>1.046 (81,53%)</b>	<b>973</b>
Haken	Standard $bp3^{TM}$	5.891	974	0 (0,0%)	12
	Direkter Transfer	5.927	908	338 (37,23%)	251
	Domänentransfer	6.124	1.065	<b>814 (76,43%)</b>	<b>475</b>

bezeichnet. Das ansichtenbasierte Verfahren zur Objektlageerkennung und die heuristische Greifplanung (Spenrath 2022) der Software werden eingesetzt. Während der Dauertests wurden Versuchsergebnisse manuell aufgezeichnet. Die Ergebnisse sind in den Tabellen 5.13 und 5.14 dargestellt.

In der Validierung des Gesamtsystems für die Rundstahlbügel wurden für den Standard  $bp3^{TM}$ -Ansatz 6.311 Werkstücke mit 1.566 Verhakungen gegriffen. Da keine Verhakungserkennung im Standard  $bp3^{TM}$  stattfindet, konnten keine Verhakungen erkannt werden. Durch den standardmäßigen Entnahmepfad wurden 170 Verhakungen unbeabsichtigt gelöst. Somit wurden 1.396 blinde Passagiere festgestellt und die Greiferfolgsrate mit 77,88% berechnet. Mit maximal eingestellter Geschwindigkeit des TCP des Robotergreifers von einem Meter pro Sekunde wurden 197 erfolgreiche Vereinzelungen pro Stunde erzielt. Der direkte Transfer schneidet mit einer Greiferfolgsrate von 83,79% und 213 erfolgreichen Vereinzelungen pro Stunde deutlich besser ab. Das beste Ergebnis liefert der Domänentransfer mit einer Greiferfolgsrate von 94,21% und 222 erfolgreichen Vereinzelungen pro Stunde.

Tabelle 5.14: Validierung des Griff-in-die-Kiste-Gesamtsystems ohne Verhakungserkennung und Enthakung als Standard  $bp3^{TM}$  und im Domänen- und direkten Transfer mit Verhakungserkennung und Enthakung für Rundstahlbügel, Pleuel und Haken in der realen Roboterzelle.

Werkstück	Ansatz	Anzahl Greifversuche	Anzahl blinde Passagiere	Greiferfolgsrate / %	Anzahl erfolgreiche Vereinzelungen pro Stunde
Rundstahl- bügel	Standard $bp3^{TM}$	6.311	1.396	77,88	197
	Direkt Transfer	5.429	880	83,79	213
	Domänentransfer	6.299	<b>365</b>	<b>94,21</b>	<b>222</b>
Pleuel	Standard $bp3^{TM}$	5.985	956	84,03	206
	Direkt Transfer	6.034	703	88,35	213
	Domänentransfer	6.089	<b>310</b>	<b>94,91</b>	<b>226</b>
Haken	Standard $bp3^{TM}$	5.891	962	83,67	206
	Direkt Transfer	5.927	657	88,92	212
	Domänentransfer	6.124	<b>590</b>	<b>90,37</b>	<b>216</b>

Bei den Pleueln schneidet der Domänentransfer bei 6.089 Werkstücken mit 1.283 Verhakungen ebenfalls am besten ab. Insgesamt werden 1.046 Verhakungen richtig erkannt und 973 davon gelöst. Dies führt zu einer Enthakungsrate von 94,91% und damit zu einer Steigerung um 10,88% gegenüber dem Standard  $bp3^{TM}$ -Ansatz. 226 Pleuel können in einer Stunde erfolgreich vereinzelt werden.

Mit einer Enthakungsrate von 90,37% liefert der Domänentransfer bei den Haken ebenfalls das beste Ergebnis. Von 6.124 Werkstücken mit 1.065 Verhakungen wurden 814 richtig erkannt und 475 erfolgreich gelöst. Pro Stunde können mit diesem Ansatz 216 Haken erfolgreich vereinzelt werden. Damit steigert der Domänentransferansatz den Standard  $bp3^{TM}$ -Ansatz um 10 Vereinzelungen pro Stunde. Bei den Haken wird in allen Versuchen zudem deutlich, dass es sich dabei um die schwierigste Geometrie der verglichenen Werkstücke handelt, da offene und geschlossene Geometriemerkmale vorhanden sind.

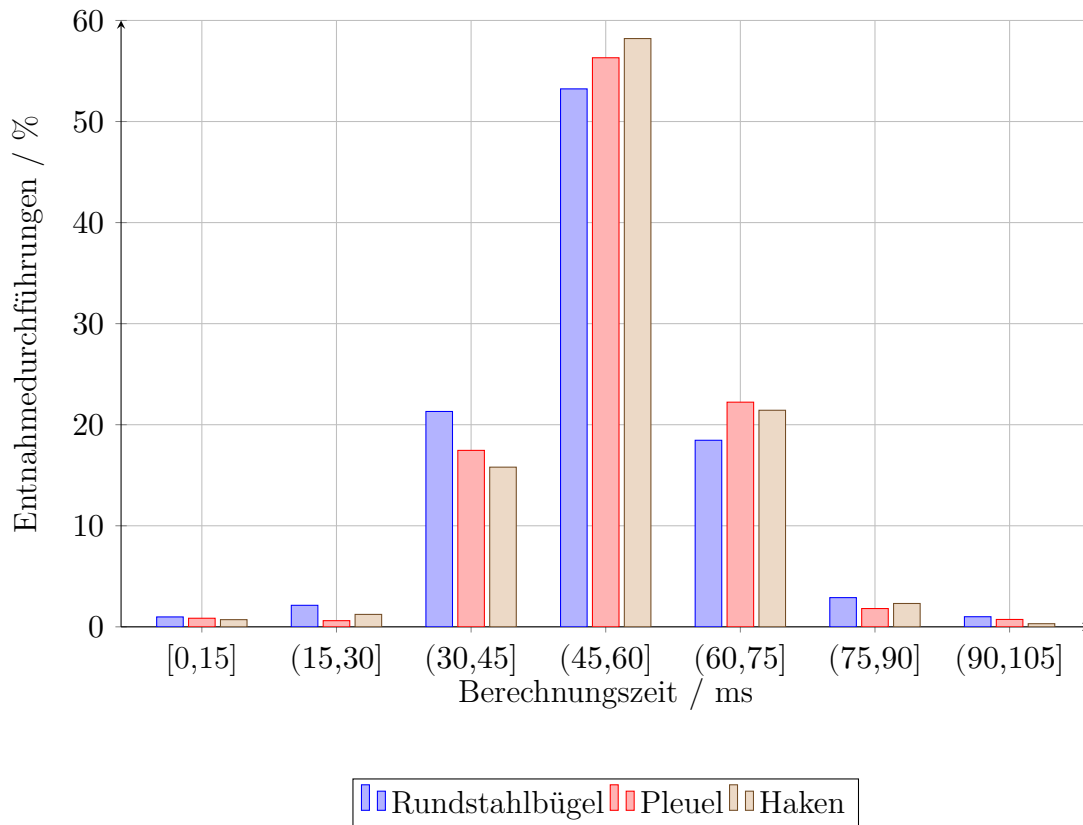


Abbildung 5.3: Histogramm der Berechnungszeit der Verhakungserkennung und Enthakung relativ zu den Entnahmedurchführungen

Die durchschnittliche Berechnungszeit des Moduls für die Verhakungserkennung und Enthakung beträgt über alle Werkstückgeometrien mit einer unterschiedlichen Anzahl an verhakten Werkstücksituationen hinweg durchschnittlich 54,13 Millisekunden. Die durchschnittliche Berechnungszeit beeinflusst die Taktzeit, welche sich für einen gesamten Griff-in-die-Kiste-Prozess bei ungefähr zehn Sekunden einpendelt, nur sehr gering. Zudem ist meist die Bewegung des Roboters die zeitlimitierende Komponente, da die Berechnungen parallel ausgeführt werden können. Die Berechnungen wurden auf einem Intel Core i9-10980XE mit 4,6GHz und 64GB Arbeitsspeicher durchgeführt. Die genaue Verteilung der Berechnungszeiten ist in dem Histogramm in Abbildung 5.3 dargestellt.

Die Ergebnisse zeigen, dass eine Steigerung der Effizienz des Gesamtsystems erzielt werden konnte. Die erfolgreichen Vereinzlungen pro Stunde konnten um bis zu 12,69% gesteigert werden. Die Anzahl der blinden Passagiere wurde bei Rundstahlbügeln um 73,85%, bei

---

Pleueln um 67,57% und bei Haken um 38,67% reduziert. Die Einrichtzeit für neue Werkstücke beträgt ungefähr 180 Stunden und kann durch wenige, manuelle Eingriffe nahezu autonom ausgeführt werden.



# 6 Zusammenfassung und Ausblick

## 6.1 Zusammenfassung

In dieser Arbeit wurde eine auf maschinellen Lernverfahren basierende Methode zur Verhakungserkennung und zur Enthakungspfadplanung für verhakende Werkstücke beim Griff-in-die-Kiste vorgestellt. Diese Methoden wurden in ein Modul integriert, welches in Griff-in-die-Kiste-Applikationen implementiert werden kann. Das Modul soll dem Griff-in-die-Kiste-System ermöglichen, Verhakungen bereits vor dem Greifen zu erkennen und damit unverhakte Werkstücke präferiert greifen zu können. Zudem soll das Modul fähig sein, verhakte Werkstücke durch zusätzliche Pfadposen im Entnahmepfad des Roboters intelligent und effizient zu trennen. Der Fokus der Entwicklung liegt auf einer hohen Effizienz, Skalierbarkeit und Flexibilität. Die Methoden sollen die effektive Taktzeit eines Griff-in-die-Kiste-Systems verkürzen und Stillstandszeiten reduzieren. Eine hohe Skalierbarkeit soll erreicht werden, indem das Modul auf neue Werkstückgeometrien adaptiert werden kann. Die Entwicklung, in Anbetracht einer hohen Flexibilität, ermöglicht es den Methoden, mit unterschiedlicher Peripherie einer Roboterzelle umgehen zu können.

Hierfür wurde zunächst eine Methode zur Verhakungserkennung einer verhakten Werkstücksituation in einer Kiste entwickelt. Durch Zuschneiden des Tiefenbildes anhand der gefundenen Werkstücklage kann eine Prognose der Verhakungswahrscheinlichkeit des Werkstücks durch ein künstliches, neuronales Netzwerk erfolgen. Das neuronale Netzwerk wurde mit synthetischen, in der Simulation generierten, Tiefenkartenzuschnitten trainiert, welche vorab durch Domänentransfermethoden der Realität angenähert wurden. Die Prognosen des neuronalen Netzwerks werden in einem Kostenfaktor der heuristischen Greifplanung zur Verfügung gestellt und somit wird das Greifen unverhakter Werkstücke präferiert. Die Verhakungserkennung wurde mit drei unterschiedlichen Werkstückgeometrien in der realen Roboterzelle getestet und erreicht bei den Rundstahlbügeln eine Verhakungserkennungsrate von 78,60%, bei den Pleueln von 81,53% und bei den Haken von 76,43%.

Um verhakte Werkstücksituationen zu lösen, wurde eine Enthakungspfadplanung entwickelt. Hierfür wurde zunächst ein Hemisphärenmodell entwickelt, welches aus drei Hemisphären und jeweils 17 Pfadposen auf der Hemisphärenoberfläche besteht. Diese Pfadposen werden mit einer Funktion bewertet, welche die Verschiebungen der Werkstücke zueinander betrachtet, und ein möglicher Enthakungspfad geplant. Neben zwei überwachten, maschinellen Lernverfahren wurde auch ein bestärkendes Lernverfahren vorgestellt und ein Vergleich der Lernverfahren durchgeführt. Durch den Domänentransfer von den simulierten Trainingsdaten und -umgebungen in die reale Roboterzelle und zusätzliche Optimierungen der Enthakungsmethode wurde für die Rundstahlbügel eine Enthakungsrate von 98,20%, für die Pleuel von 93,40% und für die Haken von 44,60% erreicht.

Das Modul, bestehend aus der vorgestellten Verhakungserkennungs- und Enthakungsmethode, kann innerhalb von 180 Stunden auf eine neue Werkstückgeometrie angepasst werden. Dieser Einrichtprozess kann unabhängig von dem Betrieb einer Griff-in-die-Kiste-Roboterzelle durchgeführt werden und bedarf geringem manuellem Einrichtaufwand. Beide Methoden sind flexibel für unterschiedliche 3D-Bildsensoren, Industrieroboter, Greifprinzipien, Greifergeometrien, Kistenabmessungen und Werkstückgeometrien einsetzbar. Zudem kann das Modul in sämtlichen Griff-in-die-Kiste-Anwendungen eingesetzt werden, die die benötigten Eingangsdaten zur Verfügung stellen. Bei einem experimentellen Einsatz des Moduls in einer realen Roboterzelle konnte bei einer durchschnittlichen Berechnungszeit des Moduls von 54,13 Millisekunden die Greiferfolgsrate bei den Rundstahlbügeln um 16,33 Prozentpunkte, bei den Pleueln um 10,88 Prozentpunkte und bei den Haken um 6,70 Prozentpunkte erhöht werden. Zudem wurde die Anzahl der blinden Passagiere bei Rundstahlbügeln um 73,85 Prozentpunkte, bei Pleueln um 67,57 Prozentpunkte und bei Haken um 38,67 Prozentpunkte reduziert.

## 6.2 Ausblick

Die entstandenen Methoden zur Verhakungserkennung und Enthakungspfadplanung können zukünftig in verschiedenen Anwendungsgebieten angewandt und weiterentwickelt werden. Die zusätzlichen Orientierungsänderungen an den Hemisphärenpfadposen sind in



dieser Arbeit durch Rotationsmodelle vorgegeben. Diese können in weiteren Entwicklungen variabel gestaltet werden und somit den Aktionsraum des Agenten für das bestärkende Lernverfahren erweitern.

Ferner kann untersucht werden, wie die Methoden der Verhakungserkennung und Enthakungspfadplanung, beispielsweise durch Meta-Lernverfahren, aufwandsärmer für neue Werkstückgeometrien generalisiert werden können. Anhand einer Taxonomie können Gemeinsamkeiten der Werkstücke herausgefunden und diese Erkenntnisse in ein Meta-Training integriert werden.

Das Modul kann mit dem Einsatz von *Visual Servoing* weiterentwickelt werden, um in Echtzeit auf Gegebenheiten in der Kistensituation zu reagieren und den Enthakungspfad noch individueller und effizienter durchzuführen und zu planen. Das Modul bietet hierfür in seiner Kommunikation mit der Greifplanung geeignete Möglichkeiten.

Eine weitere Möglichkeit des Einsatzes des Moduls ergibt sich in der Beschichtungs- und Lackierindustrie, in welcher oftmals Werkstücke in Förderstrecken ein- und abgehängt werden müssen. Die Förderstrecken sind meist mit Haken versehen, um die Werkstücke robust bewegen zu können. Mit zusätzlichen Technologien des *Visual Servoings* kann das Modul weiterentwickelt werden und somit Echtzeitdaten des Prozesses aufgenommen und verarbeitet werden.

In einer Griff-in-die-Kiste-Anwendung kann das Modul hinsichtlich weiterer Werkstückgeometrien weiterentwickelt werden. Beispielsweise können sich in einer Kiste gewollt oder unabsichtlich verschiedene Werkstückgeometrien befinden, welche nicht nur lokalisiert und gegriffen, sondern auch enthakt werden müssen. Zudem stellen biegeschlaffe Werkstücke für eine modellbasierte Vereinzelungslösung eine Schwierigkeit dar und die Simulationsumgebungen können für diese Werkstückeigenschaften erweitert werden. Bestimmte Werkstückgeometrien können sich nicht nur formschlüssig, sondern auch kraftschlüssig verhaken und verklemmen. Hierfür können die entwickelten Ansätze durch Kraftregelungen erweitert und damit die Anwendungen noch robuster gestaltet werden.



# Literatur

- Aleotti & Caselli 2008** Aleotti, Jacopo; Caselli, Stefano, 2008.  
Grasp Programming by Demonstration: A task-based quality measure.  
In: *RO-MAN 2008 - 17th IEEE International Symposium on Robot and Human Interactive Communication, Technische Universität München, Deutschland, 1.-3. August 2008*, S. 383–388.  
DOI: 10.1109/ROMAN.2008.4600696
- Argall et al. 2009** Argall, Brenna D.; Chernova, Sonia; Veloso, Manuela; Browning, Brett, 2009.  
A survey of robot learning from demonstration.  
*Robotics and Autonomous Systems* **57** (5), 2009, S. 469–483.  
DOI: 10.1016/j.robot.2008.10.024
- Berenson & Srinivasa 2008** Berenson, Dmitry; Srinivasa, Siddhartha S., 2008.  
Grasp synthesis in cluttered environments for dexterous hands.  
In: *Humanoids 2008 - 8th IEEE-RAS International Conference on Humanoid Robots, Daejeon, Korea (South)*, S. 189–196.  
DOI: 10.1109/ICHR.2008.4755944
- Berger et al. 2000** Berger, Martin; Bachler, Gernot; Scherer, Stefan, 2000.  
Vision guided bin picking and mounting in a flexible assembly cell.  
In: *Intelligent Problem Solving. Methodologies and Approaches: 13th International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems, IEA/AIE 2000 New Orleans, Louisiana, USA, June 19–22, 2000 Proceedings 13*, S. 109–117.  
ISBN: 3540676899
- Berscheid et al. 2019** Berscheid, Lars; Meißner, Pascal; Kröger, Torsten, 2019.  
Robot learning of shifting objects for grasping in cluttered environments.  
In: *2019 IEEE/RSJ international conference on intelligent robots and systems (IROS), Macau, China*, S. 612–618.  
ISBN: 1728140048
- Bohg & Kragic 2010** Bohg, Jeannette; Kragic, Danica, 2010.  
Learning grasping points with shape context.  
*Robotics and Autonomous Systems* **58** (4), 2010, S. 362–377

- Bone et al. 2008** Bone, Gary M.; Lambert, Andrew; Edwards, Mark, 2008. Automated modeling and robotic grasping of unknown three-dimensional objects.  
In: *2008 IEEE International Conference on Robotics and Automation, Pasadena, California*,  
S. 292–298.  
ISBN: 1424416469
- Bormann et al. 2019** Bormann, Richard; de Brito, Bruno Ferreira; Lindermayr, Jochen; Omainka, Marco; Patel, Mayank, 2019. Towards automated order picking robots for warehouses and retail.  
In: *Computer Vision Systems: 12th International Conference, ICVS 2019, Thessaloniki, Greece, September 23–25, 2019, Proceedings 12*,  
S. 185–198.  
ISBN: 3030349942
- Brown & Sandholm 2017** Brown, Noam; Sandholm, Tuomas, 2017. Libratus: The Superhuman AI for No-Limit Poker.  
In: *Proceedings of the 26th International Joint Conference on Artificial Intelligence, Melbourne, Australia*,  
S. 5226–5228.  
ISBN: 9780999241103
- Chen et al. 2018** Chen, Yu-Kai; Sun, Guo-Jhen; Lin, Huei-Yung; Chen, Shyh-Leh, 2018. Random bin picking with multi-view image acquisition and CAD-based pose estimation.  
In: *2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Miyazaki, Japan*,  
S. 2218–2223.  
ISBN: 1538666502
- Claus & Schwill 2001** Claus, Volker; Schwill, Andreas, 2001. *Duden Informatik: ein Fachlexikon für Studium und Praxis*. Mannheim: Dudenverlag.  
ISBN: 978-3411100231
- Craig 2006** Craig, John J, 2006. *Introduction to robotics: Mechanics and control*. London: Pearson Higher Education.  
ISBN: 9780131236295
- Czichos 2015** Czichos, Horst, 2015. *Mechatronik: Grundlagen und Anwendungen technischer Systeme*. Wiesbaden: Springer Vieweg.  
ISBN: 365809950X

- Denavit & Hartenberg 1955** Denavit, Jacques; Hartenberg, Richard S., 1955.  
A Kinematic Notation for Lower-Pair Mechanisms Based on Matrices.  
*Journal of Applied Mechanics, Transactions ASME* **22** (2), 1955, S. 215–221
- Deng et al. 2016** Deng, Yue; Bao, Feng; Kong, Youyong; Ren, Zhiqian; Dai, Qionghai, 2016.  
Deep direct reinforcement learning for financial signal representation and trading.  
*IEEE transactions on neural networks and learning systems* **28** (3), 2016, S. 653–664
- Dessimoz et al. 1984** Dessimoz, Jean-Daniel; Birk, John R.; Kelley, Robert B.; Martins, Henrique A. S.; Lin, Chi, 1984.  
Matched filters for bin picking.  
*IEEE transactions on pattern analysis and machine intelligence* (6), 1984, S. 686–697
- Domae et al. 2014** Domae, Yukiyasu; Okuda, Haruhisa; Taguchi, Yuichi; Sumi, Kazuhiko; Hirai, Takashi, 2014.  
Fast graspability evaluation on single depth maps for bin picking with general grippers.  
In: *IEEE International Conference on Robotics and Automation (ICRA), Hong Kong, China*, S. 1997–2004.  
ISBN: 1479936855
- Dong et al. 2020** Dong, Hao; Ding, Zihan; Zhang, Shanghang, 2020.  
*Deep Reinforcement Learning*. Singapore: Springer Singapore. ISBN: 978-981-15-4094-3
- Dong et al. 2019** Dong, Zhipeng; Wang, Mi; Wang, Yanli; Zhu, Ying; Zhang, Zhiqi, 2019.  
Object detection in high resolution remote sensing imagery based on convolutional neural networks with suitable object scale features.  
*IEEE Transactions on Geoscience and Remote Sensing* **58** (3), 2019, S. 2104–2114
- Dune et al. 2008** Dune, Claire; Remazeilles, Anthony; Marchand, Eric; Leroux, Cédric, 2008.  
Vision-based grasping of unknown objects to improve disabled people autonomy.  
In: *Robotics: Science and Systems Manipulation Workshop: Intelligence in Human Environments*.
- Dupuis et al. 2008** Dupuis, Donna C; Léonard, Simon; Baumann, Matthew A; Croft, Elizabeth A; Little, James J, 2008.  
Two-fingered grasp planning for randomized bin-picking.  
In: *Robotics: Science and Systems Manipulation Workshop: Intelligence in Human Environments*.

- Effenberger et al. 2013** Effenberger, Ira; Kühnle, Jens; Verl, Alexander, 2013.  
Fast and flexible 3D object recognition solutions for machine vision applications.  
In: Bingham, Philip R.; Lam, Edmund Y. (Hrsg.): *Image Processing: Machine Vision Applications VI, Proc. SPIE 8661*, S. 191–200.  
DOI: 10.1117/12.2006990
- Ekvall & Kragic 2004** Ekvall, Staffan; Kragic, Danica, 2004.  
Interactive grasp learning based on human demonstration.  
In: *IEEE International Conference on Robotics and Automation (ICRA), 2004, New Orleans, LA, USA*, S. 3519–3524.  
ISBN: 0780382323
- Ferrari & Canny 1992** Ferrari, Carlo; Canny, John F, 1992.  
Planning optimal grasps.  
In: *IEEE International Conference on Robotics and Automation (ICRA) 1992, Nice, France*, S. 6
- Ford & Crowther 1922** Ford, Henry; Crowther, Samuel, 1922.  
*My life and work: In collaboration with Samuel Crowther*.  
Cornstalk Publishing Company.  
ISBN: 978-8184682120
- François-Lavet et al. 2018** François-Lavet, Vincent; Henderson, Peter; Islam, Riashat; Bellemare, Marc G.; Pineau, Joelle, 2018.  
An introduction to deep reinforcement learning.  
*Foundations and Trends in Machine Learning* **11** (3-4), 2018, S. 219–354.  
DOI: 10.1561/22000000071
- Frochte 2021** Frochte, Jörg, 2021.  
*Maschinelles Lernen: Grundlagen und Algorithmen in Python*.  
3., überarbeitete und erweiterte Auflage. München: Hanser.  
Plus Hanser-Fachbuch.  
ISBN: 9783446461444
- Gandhi et al. 2017** Gandhi, Dhiraj; Pinto, Lerrel; Gupta, Abhinav, 2017.  
Learning to fly by crashing.  
In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vancouver, Canada*, S. 3948–3955.  
ISBN: 1538626829
- Glassner 1989** Glassner, Andrew S., 1989.  
*An introduction to ray tracing*. Morgan Kaufmann.  
ISBN: 0122861604

- Goldfeder et al. 2007** Goldfeder, Corey; Allen, Peter K; Lackner, Claire; Pelosof, Raphael, 2007.  
Grasp planning via decomposition trees.  
In: *IEEE International Conference on Robotics and Automation (ICRA)*, Rom, Italien, S. 4679–4684.  
DOI: 10.1109/ROBOT.2007.364200
- Goodfellow et al. 2016** Goodfellow, Ian; Bengio, Yoshua; Courville, Aaron, 2016.  
*Deep Learning*. MIT press.  
ISBN: 0262337371
- Harada et al. 2013** Harada, Kensuke; Nagata, Kazuyuki; Tsuji, Tokuo; Yamanobe, Natsuki; Nakamura, Akira; Kawai, Yoshihiro, 2013.  
Probabilistic approach for object bin picking approximated by cylinders.  
In: *IEEE International Conference on Robotics and Automation (ICRA)*, Karlsruhe, Deutschland, S. 3742–3747.  
DOI: 10.1109/ICRA.2013.6631103
- Hartenberg & Denavit 1964** Hartenberg, Richard; Denavit, Jacques, 1964.  
*Kinematic synthesis of linkages*. New York: McGraw-Hill.  
ISBN: 978-0-07-026910-1
- He et al. 2016** He, Kaiming; Zhang, Xiangyu; Ren, Shaoqing; Sun, Jian, 2016.  
Deep Residual Learning for Image Recognition.  
In: *IEEE conference on computer vision and pattern recognition (CVPR)*, Las Vegas, NV, USA, S. 770–778.  
URL: [http://www.cv-foundation.org/openaccess/content\\_cvpr\\_2016/](http://www.cv-foundation.org/openaccess/content_cvpr_2016/)  
Zugriff am: 16.06.2023
- He et al. 2021** He, Tierui; Aslam, Shoaib; Tong, Zhekai; Seo, Jungwon, 2021.  
Scooping Manipulation via Motion Control with a Two-fingered Gripper and its Application to Bin Picking.  
*IEEE Robotics and Automation Letters (RAL)* **6** (4), 2021, S. 6394–6401.  
DOI: 10.1109/LRA.2021.3093896
- Heiß 1985** Heiß, Hermann, 1985.  
*Die explizite Lösung der kinematischen Gleichung für eine Klasse von Industrierobotern*. Berlin, Technische Universität, Dissertation, 1985. 34. Jahrgang, Heft 7.  
DOI: 10.1524/auto.1986.34.112.296
- Heiß 1986** Heiß, Hermann, 1986.  
*Konstruktionskriterien und Lösungsverfahren für Industrieroboter mit explizit lösbarer kinematischer Gleichung*. Springer, Berlin, Heidelberg, New York, NY. 3

- Horn & Ikeuchi 1983** Horn, Berthold K. P.; Ikeuchi, Katsushi, 1983.  
Picking parts out of a bin.  
*Massachusetts Institute of Technology Cambridge Artificial Intelligence Lab*, 1983.  
URL: <https://apps.dtic.mil/sti/citations/ADA139257>  
Zugriff am: 16.06.2023
- Horn & Ikeuchi 1984** Horn, Berthold K. P.; Ikeuchi, Katsushi, 1984.  
The Mechanical Manipulation of Randomly Oriented Parts.  
*Scientific American* **251** (2), 1984, S. 100–113.  
URL: <http://www.jstor.org/stable/24969439>  
Zugriff am: 16.06.2023
- Huang et al. 2017** Huang, Gao; Liu, Zhuang; van der Maaten, Laurens; Weinberger, Kilian Q., 2017.  
Densely Connected Convolutional Networks.  
In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, Hawaii*, S. 4700–4708.  
URL: [https://openaccess.thecvf.com/content\\_cvpr\\_2017/html/Huang\\_Densely\\_Connected\\_Convolutional\\_CVPR\\_2017\\_paper.html](https://openaccess.thecvf.com/content_cvpr_2017/html/Huang_Densely_Connected_Convolutional_CVPR_2017_paper.html)  
Zugriff am: 16.06.2023
- Huebner & Kragic 2008** Huebner, Kai; Kragic, Danica, 2008.  
Selection of robot pre-grasps using box-based shape approximation.  
In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Nizza, Frankreich*, S. 1765–1770.  
ISBN: 1424420571
- Ikeuchi et al. 1983** Ikeuchi, Katsushi; Horn, Berthold K. P.; Nagata, Shigemi; Callahan, Tom; Feingold, Oded, 1983.  
Picking up an Object from a Pile of Objects.  
*Massachusetts Institute of Technology Cambridge Artificial Intelligence Lab*, 1983.  
URL: <https://apps.dtic.mil/sti/citations/ADA133631>  
Zugriff am: 16.06.2023
- ISO 8373** ISO 8373:2012.  
*Robots and Robotic Devices*
- Johns et al. 2016** Johns, Edward; Leutenegger, Stefan; Davison, Andrew J., 2016.  
Deep Learning a Grasp Function for Grasping under Gripper Pose Uncertainty.  
In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Daejeon, Korea*, S. 4461–4468.  
ISBN: 1509037624



- Kalashnikov et al. 2018** Kalashnikov, Dmitry; Irpan, Alex; Pastor, Peter; Ibarz, Julian; Herzog, Alexander; Jang, Eric; Quillen, Deirdre; Holly, Ethan; Kalakrishnan, Mrinal; Vanhoucke, Vincent, 2018.  
Qt-opt: Scalable deep reinforcement learning for vision-based robotic manipulation.  
*arXiv preprint arXiv:1806.10293*, 2018
- Khalid et al. 2021** Khalid, Muhammad Usman; Spenrath, Felix; Mönnig, Manuel; Moosmann, Marius; Bormann, Richard; Kunz, Holger; Huber, Marco F., 2021.  
Automatic Grasp Generation for Vacuum Grippers for Random Bin Picking.  
In: *Advances in Automotive Production Technology—Theory and Application: Stuttgart Conference on Automotive Production (SCAP), Stuttgart, Deutschland*, S. 247–255.  
ISBN: 3662629615
- El-Khoury & Sahbani 2010** El-Khoury, Sahar; Sahbani, Anis, 2010.  
A new strategy combining empirical and analytical approaches for grasping unknown 3D objects.  
*Robotics and Autonomous Systems* **58** (5), 2010, S. 497–507.  
DOI: 10.1016/j.robot.2010.01.008
- Kimitoshi et al. 2010** Kimitoshi, Yamazaki; Tomono, Masahiro; Tsubouchi, Takashi; Jimenez, A; Hadithi, BMA, 2010.  
*Robot Manipulator Trends and Development*. Autonomous 3D shape modeling and grasp planning for handling unknown objects. InTech.  
ISBN: 978-953-307-073-5
- Kleeberger & Huber 2020** Kleeberger, Kilian; Huber, Marco F., 2020.  
Single Shot 6D Object Pose Estimation.  
In: *IEEE International Conference on Robotics and Automation (ICRA), virtuell*, S. 6239–6245.  
ISBN: 1728173957
- Kleeberger et al. 2021** Kleeberger, Kilian; Schnitzler, Jonathan; Khalid, Muhammad Usman; Bormann, Richard; Kraus, Werner; Huber, Marco F., 2021.  
Precise Object Placement with Pose Distance Estimations for Different Objects and Grippers.  
In: *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, S. 4639–4646.  
ISBN: 1665417145

- Konishi et al. 2019** Konishi, Yoshinori; Hattori, Kosuke; Hashimoto, Manabu, 2019.  
Real-time 6D Object Pose Estimation on CPU.  
In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Macau, China*,  
S. 3451–3458.  
ISBN: 1728140048
- Krizhevsky et al. 2017** Krizhevsky, Alex; Sutskever, Ilya; Hinton, Geoffrey E, 2017.  
Imagenet Classification with Deep Convolutional Neural Networks.  
*Communications of the ACM* **60** (6), 2017, S. 84–90.  
DOI: 10.1145/3065386
- Kunz et al. 2015** Kunz, Holger; Löchte, Christian; Dietrich, Franz;  
Raatz, Annika; Fischer, Fabian; Dröder, Klaus; Dilger, Klaus, 2015.  
Novel form-flexible handling and joining tool for automated preforming.  
*Science and Engineering of Composite Materials* **22** (2), 2015, S. 199–213.  
DOI: 10.1515/secm-2013-0326
- LaValle & Kuffner Jr 2001** LaValle, Steven M; Kuffner Jr, James J, 2001.  
Randomized kinodynamic planning.  
*The international journal of robotics research* **20** (5), 2001, S. 378–400.  
URL: <https://journals.sagepub.com/toc/ijra/20/5>  
Zugriff am: 16.06.2023
- Leão et al. 2020** Leão, Gonçalo; Costa, Carlos M.; Sousa, Armando;  
Veiga, Germano, 2020.  
Detecting and solving tube entanglement in bin picking operations.  
*Applied Sciences* **10** (7), 2020, S. 2264.  
DOI: 10.3390/app10072264
- Leão et al. 2022** Leão, Gonçalo; Camacho, Rui; Sousa, Armando;  
Veiga, Germano, 2022.  
An Inductive Logic Programming Approach for Entangled Tube Modeling in Bin Picking.  
In: *ROBOT2022: Fifth Iberian Robotics Conference: Advances in Robotics, Volume 2*,  
S. 79–91.  
DOI: 10.1007/978-3-031-21062-4\_7
- LeCun 1989** LeCun, Yann, 1989.  
Generalization and Network Design Strategies.  
In: Pfeifer, R.; Schreter, Z.; Fogelman, F.; Steels, L. (Hrsg.):  
*Connectionism in perspective, Elsevier, Zürich, Schweiz*,  
S. 18

- Ledermann 2012** Ledermann, Thomas, 2012.  
*Partikel-Schwarm-Optimierung zur Objektlageerkennung in Tiefendaten*. Stuttgart, Universität, Dissertation, 2012.  
ISBN: 3939890979
- Lenz et al. 2015** Lenz, Ian; Lee, Honglak; Saxena, Ashutosh, 2015.  
Deep Learning for Detecting Robotic Grasps.  
*The International Journal of Robotics Research* **34** (4-5),  
2015, S. 705–724.  
DOI: 10.1177/0278364914549607
- Levine et al. 2016** Levine, Sergey; Finn, Chelsea; Darrell, Trevor;  
Abbeel, Pieter, 2016.  
End-to-end Training of Deep Visuomotor Policies.  
*The Journal of Machine Learning Research* **17** (1), 2016, S.  
1334–1373.  
URL:  
<https://www.jmlr.org/papers/volume17/15-522/15-522.pdf>  
Zugriff am: 16.06.2023
- Levine et al. 2018** Levine, Sergey; Pastor, Peter; Krizhevsky, Alex; Ibarz, Julian;  
Quillen, Deirdre, 2018.  
Learning hand-eye coordination for robotic grasping with  
deep learning and large-scale data collection.  
*The International journal of robotics research* **37** (4-5), 2018,  
S. 421–436.  
DOI: 10.1177/0278364917710318
- Lin 1993** Lin, L.-J., 1993.  
Hierarchical Learning of Robot Skills by Reinforcement.  
In: *IEEE International Conference on Neural Networks*,  
181–186, Vol. 1.  
DOI: 10.1109/ICNN.1993.298553
- Lin et al. 1997** Lin, Ming C; Manocha, Dinesh; Cohen, Jon;  
Gottschalk, Stefan, 1997.  
Collision Detection: Algorithms and Applications.  
*Algorithms for robotic motion and manipulation*, 1997, S.  
129–142.  
ISBN: 978-1-56881-067-6
- Lin & Sun 2015** Lin, Yun; Sun, Yu, 2015.  
Robot Grasp Planning based on Demonstrated Grasp  
Strategies.  
*The International Journal of Robotics Research* **34** (1), 2015,  
S. 26–42.  
DOI: 10.1177/0278364914555544

- Löchte et al. 2014** Löchte, Christian; Kunz, Holger; Schnurr, Raphael; Langhorst, Sören; Dietrich, Franz; Raatz, Annika; Dilger, Klaus; Dröder, Klaus, 2014. Form-Flexible Handling and Joining Technology (FormHand) for the Forming and Assembly of Limp Materials. *Procedia CIRP* **23**, 2014, S. 206–211. 5th CIRP Conference on Assembly Technologies and Systems (CATS), Dresden, Deutschland. DOI: 10.1016/j.procir.2014.10.086
- Luo et al. 2015** Luo, Ren C.; Kuo, Chia-Wen; Chung, Yi-Ting, 2015. Model-based 3D object recognition and fetching by a 7-DoF robot with online obstacle avoidance for factory automation. In: *IEEE International Conference on Robotics and Automation (ICRA), Seattle, WA, USA*, S. 2647–2652. DOI: 10.1109/ICRA.2015.7139556
- Mahler et al. 2017** Mahler, Jeffrey; Liang, Jacky; Niyaz, Sherdil; Laskey, Michael; Doan, Richard; Liu, Xinyu; Ojea, Juan Aparicio; Goldberg, Ken, 2017. Dex-net 2.0: Deep learning to plan robust grasps with synthetic point clouds and analytic grasp metrics. *arXiv preprint arXiv:1703.09312*, 2017
- Martinez et al. 2015** Martinez, Carlos; Boca, Remus; Zhang, Biao; Chen, Heping; Nidamarthi, Srinivas, 2015. Automated bin picking system for randomly located industrial parts. In: *IEEE International Conference on Technologies for Practical Robot Applications (TePRA), Woburn, MA, USA*, S. 1–6. ISBN: 1479987573
- Matsumura et al. 2019** Matsumura, Ryo; Domae, Yukiyasu; Wan, Weiwei; Harada, Kensuke, 2019. Learning based robotic bin-picking for potentially tangled objects. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Macau, China*, S. 7990–7997. ISBN: 1728140048
- McCarthy et al. 1955** McCarthy, John; Minsky, Marvin L.; Rochester, Nathaniel; Shannon, Claude E., 1955. A Proposal for the Dartmouth Summer Research Project on Artificial Intelligence. *AI Magazine* **27** (4), 1955, S. 12. DOI: 10.1609/aimag.v27i4.1904

- Miller et al. 2003** Miller, A.T.; Knoop, S.; Christensen, H.I.; Allen, P.K., 2003. Automatic Grasp Planning using Shape Primitives. In: *IEEE International Conference on Robotics and Automation (Cat. No.03CH37422) (ICRA), Taipei, Taiwan*, 1824–1829 vol.2. DOI: 10.1109/ROBOT.2003.1241860
- Mitchell 1997** Mitchell, Tom M., 1997. Does Machine Learning Really Work? *AI Magazine* **18** (3), 1997, S. 11. DOI: 10.1609/aimag.v18i3.1303
- Mnih et al. 2013** Mnih, Volodymyr; Kavukcuoglu, Koray; Silver, David; Graves, Alex; Antonoglou, Ioannis; Wierstra, Daan; Riedmiller, Martin, 2013. Playing Atari with Deep Reinforcement Learning. *NIPS Deep Learning Workshop*, 2013. DOI: 10.48550/ARXIV.1312.5602
- Mnih et al. 2015** Mnih, Volodymyr; Kavukcuoglu, Koray; Silver, David; Rusu, Andrei A.; Veness, Joel; Bellemare, Marc G.; Graves, Alex; Riedmiller, Martin; Fidjeland, Andreas K.; Ostrovski, Georg, 2015. Human-level Control through Deep Reinforcement Learning. *nature* **518** (7540), 2015, S. 529–533. DOI: 10.1038/nature14236
- Mohri et al. 2018** Mohri, Mehryar; Rostamizadeh, Afshin; Talwalkar, Ameet, 2018. *Foundations of machine learning*. MIT press. ISBN: 9780262039406
- Moosmann et al. 2020** Moosmann, Marius; Spennath, Felix; Kleberger, Kilian; Khalid, Muhammad Usman; Mönnig, Manuel; Rosport, Johannes; Bormann, Richard, 2020. Increasing the Robustness of Random Bin Picking by Avoiding Grasps of Entangled Workpieces. *Procedia CIRP* **93**, 2020, S. 1212–1217. 53rd CIRP Conference on Manufacturing Systems (CMS), Chicago, IL, USA. DOI: 10.1016/j.procir.2020.03.082
- Moosmann et al. 2021a** Moosmann, Marius; Kulig, Marco; Spennath, Felix; Mönnig, Manuel; Roggendorf, Simon; Petrovic, Oliver; Bormann, Richard; Huber, Marco F., 2021. Separating Entangled Workpieces in Random Bin Picking using Deep Reinforcement Learning. *Procedia CIRP* **104**, 2021, S. 881–886. 54th CIRP Conference on Manufacturing Systems (CMS), Athen, Griechenland - Towards Digitalized Manufacturing 4.0. DOI: 10.1016/j.procir.2021.11.148

- Moosmann et al. 2021b** Moosmann, Marius; Spennath, Felix; Mönning, Manuel; Khalid, Muhammad Usman; Jaumann, Marvin; Rosport, Johannes; Bormann, Richard, 2021. Using Deep Neural Networks to Separate Entangled Workpieces in Random Bin Picking. In: *Advances in Automotive Production Technology—Theory and Application: Stuttgart Conference on Automotive Production (SCAP), Stuttgart, Deutschland*, S. 238–246. ISBN: 3662629615
- Moosmann et al. 2022a** Moosmann, Marius; Kaiser, Manuel; Rosport, Johannes; Spennath, Felix; Kraus, Werner; Bormann, Richard; Huber, Marco F., 2022. Performance Comparison of Supervised and Reinforcement Learning Approaches for Separating Entanglements in a Bin-Picking Application. In: *Advances in Automotive Production Technology—Theory and Application: Stuttgart Conference on Automotive Production (SCAP), Stuttgart, Deutschland*. DOI: 10.1007/978-3-031-27933-1\_15
- Moosmann et al. 2022b** Moosmann, Marius; Spennath, Felix; Rosport, Johannes; Melzer, Philipp; Kraus, Werner; Bormann, Richard; Huber, Marco F., 2022. Transfer Learning for Machine Learning-based Detection and Separation of Entanglements in Bin-Picking Applications. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Kyoto, Japan*, S. 1123–1130. ISBN: 1665479272
- Moravčík et al. 2017** Moravčík, Matej; Schmid, Martin; Burch, Neil; Lisý, Viliam; Morrill, Dustin; Bard, Nolan; Davis, Trevor; Waugh, Kevin; Johanson, Michael; Bowling, Michael, 2017. Deepstack: Expert-level artificial intelligence in heads-up no-limit poker. *Science (New York, N.Y.)* **356** (6337), 2017, S. 508–513. DOI: 10.1126/science.aam6960
- Morrison et al. 2018** Morrison, Douglas; Corke, Peter; Leitner, Jürgen, 2018. Closing the Loop for Robotic Grasping: A Real-time, Generative Grasp Synthesis Approach. In: *Robotics: Science and Systems (RSS), Columbia University, New York City, NY, USA*. DOI: 10.48550/ARXIV.1804.05172
- Murphy 2012** Murphy, Kevin P., 2012. *Machine learning: a probabilistic perspective*. MIT press. ISBN: 0262304325

- Neapolitan & Jiang 2018** Neapolitan, Richard E.; Jiang, Xia, 2018.  
*Artificial intelligence: With an introduction to machine learning*. CRC Press.  
ISBN: 1351384392
- Palzkill & Verl 2012** Palzkill, Matthias; Verl, Alexander, 2012.  
Object Pose Detection in Industrial Environment.  
In: *ROBOTIK 2012; 7th German Conference on Robotics, München, Deutschland*,  
S. 1–5.  
ISBN: 3800734184
- Palzkill 2014** Palzkill, Matthias, 2014.  
*Stuttgarter Beiträge zur Produktionsforschung*. Bd. 37,  
Heuristisches Suchverfahren zur Objektlageerkennung aus  
Punktwolken für industrielle Zuführsysteme. Universität  
Stuttgart, Dissertation.  
ISBN: 978-3-8396-0784-8
- Pan et al. 2017** Pan, Xinlei; You, Yurong; Wang, Ziyang; Lu, Cewu, 2017.  
Virtual to Real Reinforcement Learning for Autonomous  
Driving.  
In: *British Machine Vision Conference (BMVC), London, England*.  
DOI: 10.48550/ARXIV.1704.03952
- Papazov et al. 2012** Papazov, Chavdar; Haddadin, Sami; Parusel, Sven;  
Krieger, Kai; Burschka, Darius, 2012.  
Rigid 3D geometry matching for grasping of known objects in  
cluttered scenes.  
*The International Journal of Robotics Research* **31** (4), 2012,  
S. 538–553.  
DOI: 10.1177/0278364911436019
- Paul & Zhang 1986** Paul, Richard P; Zhang, Hong, 1986.  
Computationally efficient kinematics for manipulators with  
spherical wrists based on the homogeneous transformation  
representation.  
*The International Journal of Robotics Research* **5** (2), 1986, S.  
32–44.  
DOI: 10.1177/027836498600500204
- Pauli 1998** Pauli, Josef, 1998.  
Learning to Recognize and Grasp Objects.  
*Autonomous Robots* **5**, 1998, S. 407–420.  
DOI: 10.1023/A:1008874725911
- Pinto et al. 2017** Pinto, Lerrel; Andrychowicz, Marcin; Welinder, Peter;  
Zaremba, Wojciech; Abbeel, Pieter, 2017.  
Asymmetric Actor Critic for Image-Based Robot Learning.  
In: *14th Robotics: Science and Systems (RSS), Pittsburgh, USA*.  
DOI: 10.48550/ARXIV.1710.06542

- Pott & Dietz 2019** Pott, Andreas; Dietz, Thomas, 2019.  
*Industrielle Robotersysteme*. Berlin, Heidelberg: Springer.  
ISBN: 978-3-658-25344-8
- Ray & Howard 2020** Ray, Prabhakar; Howard, Matthew J., 2020.  
Robotic Untangling of Herbs and Salads with Parallel Grippers.  
In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Las Vegas, NV, USA*, S. 2624–2629.  
DOI: 10.1109/IROS45743.2020.9342536
- Rey & Wender 2019** Rey, Günter Daniel; Wender, Karl F., 2019.  
*Neuronale Netze: Eine Einführung in die Grundlagen, Anwendungen und Datenauswertung*. Hogrefe AG.  
ISBN: 3456957963
- Richter 2019** Richter, Stefan, 2019.  
*Statistisches und maschinelles Lernen*. Berlin, Heidelberg: Springer.  
ISBN: 978-3-662-59353-0
- Richtsfeld & Vincze 2008** Richtsfeld, Mario; Vincze, Markus, 2008.  
Grasping of Unknown Objects from a Table Top.  
In: *Workshop on Vision in Action: Efficient strategies for cognitive agents in complex environments*.  
URL: <https://inria.hal.science/inria-00325794>  
Zugriff am: 16.06.2023
- Rusu et al. 2016** Rusu, Andrei A.; Rabinowitz, Neil C.; Desjardins, Guillaume; Soyer, Hubert; Kirkpatrick, James; Kavukcuoglu, Koray; Pascanu, Razvan; Hadsell, Raia, 2016.  
Progressive Neural Networks.  
*arXiv preprint arXiv:1606.04671*, 2016.  
DOI: 10.48550/ARXIV.1606.04671
- Sanz et al. 2005** Sanz, P.J.; Requena, A.; Inesta, J.M.; Del Pobil, A.P., 2005.  
Grasping the not-so-obvious: vision-based object handling for industrial applications.  
*IEEE Robotics & Automation Magazine* **12** (3), 2005, S. 44–52.  
DOI: 10.1109/MRA.2005.1511868
- Schyja et al. 2012** Schyja, Adrian; Hypki, Alfred; Kuhlenkötter, Bernd, 2012.  
A modular and extensible framework for real and virtual bin-picking environments.  
In: *IEEE International Conference on Robotics and Automation (ICRA), St Paul, MN, USA*, S. 5246–5251.  
DOI: 10.1109/ICRA.2012.6224875



- Shao et al. 2019** Shao, Quanquan; Hu, Jie; Wang, Weiming; Fang, Yi; Liu, Wenhai; Qi, Jin; Ma, Jin, 2019. Suction Grasp Region Prediction Using Self-supervised Learning for Object Picking in Dense Clutter. In: *IEEE 5th International Conference on Mechatronics System and Robots (ICMSR), Singapore*, S. 7–12. DOI: 10.1109/ICMSR.2019.8835468
- Siciliano et al. 2008** Siciliano, Bruno; Khatib, Oussama; Kröger, Torsten, 2008. *Springer handbook of robotics*. Berlin: Springer. ISBN: 978-3-319-32550-7
- Simonyan & Zisserman 2014** Simonyan, Karen; Zisserman, Andrew, 2014. Very Deep Convolutional Networks for Large-Scale Image Recognition. In: *3rd International Conference on Learning Representations (ICLR), San Diego, CA, USA*. DOI: 10.48550/ARXIV.1409.1556
- Spennath et al. 2012** Spennath, Felix; Spiller, Alexander; Verl, Alexander, 2012. Gripping Point Determination and Collision Prevention in a Bin-Picking application. In: *ROBOTIK 2012; 7th German Conference on Robotics, München, Deutschland*, S. 1–6. ISBN: 3800734184
- Spennath et al. 2013** Spennath, Felix; Palzkill, Matthias; Pott, Andreas; Verl, Alexander, 2013. Object recognition: Bin-picking for industrial use. In: *IEEE International Conference on Intelligence and Safety for Robotics (ISR), Seoul, Korea (South)*, S. 1–3. DOI: 10.1109/ISR.2013.6695743
- Spennath & Pott 2017** Spennath, Felix; Pott, Andreas, 2017. Gripping point determination for bin picking using heuristic search. In: *10th CIRP Conference on Intelligent Computation in Manufacturing Engineering (ICME), Ischia, Italien*, S. 606–611. DOI: 10.1016/j.procir.2016.06.015
- Spennath 2022** Spennath, Felix, 2022. *Stuttgarter Beiträge zur Produktionsforschung*. Bd. 137, Heuristisches Suchverfahren für die effiziente Planung zum Greifen ungeordnet gelagerter Werkstücke mit Industrierobotern. Universität Stuttgart, Fraunhofer-Institut für Produktionstechnik und Automatisierung IPA, Dissertation. DOI: 10.18419/opus-12612

- Sutton & Barto 2018** Sutton, Richard S.; Barto, Andrew G., 2018.  
*Reinforcement Learning: An introduction*. MIT press.  
ISBN: 0262352702
- Takenouchi et al. 1998** Takenouchi, Ayako; Kanamaru, Naoyoshi; Mizukawa, Makoto, 1998.  
Hough-space-based object recognition tightly coupled with path planning for robust and fast bin-picking.  
In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Victoria, BC, Canada. Innovations in Theory, Practice and Applications (Cat. No. 98CH36190)*,  
S. 1222–1229.  
ISBN: 0780344650
- Telea 2004** Telea, Alexandru, 2004.  
An Image Inpainting Technique Based on the Fast Marching Method.  
*Journal of Graphics Tools* **9** (1), 2004, S. 23–34.  
DOI: 10.1080/10867651.2004.10487596
- Trobina 1995** Trobina, Marjan, 1995.  
*From planar patches to grasps: A 3-D robot vision system handling unmodeled objects*. 1. Aufl. Konstanz: Hartung-Gorre, Zürich, Eidgenöss. Techn. Hochschule, Dissertation.  
ISBN: 3891919735
- Van Oord et al. 2016** Van Oord, Aaron; Dieleman, Sander; Zen, Heiga; Simonyan, Karen; Vinyals, Oriol; Graves, Alex; Kalchbrenner, Nal; Senior, Andrew; Kavukcuoglu, Koray, 2016.  
WaveNet: A Generative Model for Raw Audio.  
In: *9th ISCA Speech Synthesis Workshop, Sunnyvale, CA, USA*.  
DOI: 10.48550/arXiv.1609.03499
- Vermorel & Mohri 2005** Vermorel, Joannes; Mohri, Mehryar, 2005.  
Multi-armed bandit algorithms and empirical evaluation.  
In: *Machine Learning: ECML 2005: 16th European Conference on Machine Learning, Porto, Portugal, October 3-7, 2005. Proceedings 16*,  
S. 437–448.  
ISBN: 3540292438
- Watkins 1989** Watkins, Christopher John Cornish Hellaby, 1989.  
Learning from delayed rewards,  
1989.  
URL:  
[https://www.academia.edu/download/50360235/Learning\\_from\\_delayed\\_rewards\\_20161116-28282-v2pwwq.pdf](https://www.academia.edu/download/50360235/Learning_from_delayed_rewards_20161116-28282-v2pwwq.pdf)  
Zugriff am: 16.06.2023

- Welsch et al. 2018** Welsch, Andreas; Eitle, Verena; Buxmann, Peter, 2018. Maschinelles Lernen. *HMD Praxis der Wirtschaftsinformatik* **55** (2), 2018, S. 366–382. DOI: 10.1365/s40702-018-0404-z
- World Robotics Report 2022** World Robotics Report, 2022. *International Federation of Robotics*, 2022. URL: [https://ifr.org/news/WR-2022-IR\\_PR](https://ifr.org/news/WR-2022-IR_PR) Zugriff am: 16.06.2023
- Yin et al. 2017** Yin, Wenpeng; Kann, Katharina; Yu, Mo; Schütze, Hinrich, 2017. Comparative study of CNN and RNN for natural language processing. *arXiv preprint arXiv:1702.01923*, 2017. DOI: 10.48550/arXiv.1702.01923
- You et al. 2018** You, Fang; Mende, Michael; Štogl, Denis; Hein, Björn; Kröger, Torsten, 2018. Model-free grasp planning for configurable vacuum grippers. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, Spanien*, S. 4554–4561. ISBN: 1538680947
- Zeng et al. 2020** Zeng, Andy; Song, Shuran; Lee, Johnny; Rodriguez, Alberto; Funkhouser, Thomas, 2020. TossingBot: Learning to Throw Arbitrary Objects With Residual Physics. *IEEE Transactions on Robotics* **36** (4), 2020, S. 1307–1319. DOI: 10.1109/TRO.2020.2988642
- Zeng et al. 2022** Zeng, Andy; Song, Shuran; Yu, Kuan-Ting; Donlon, Elliott; Hogan, Francois R; Bauza, Maria; Ma, Daolin; Taylor, Orion; Liu, Melody; Romo, Eudald et al., 2022. Robotic pick-and-place of novel objects in clutter with multi-affordance grasping and cross-domain image matching. *The International Journal of Robotics Research* **41** (7), 2022, S. 690–705. DOI: 10.1177/0278364919868017
- Zhang et al. 2019** Zhang, Jingqing; Lertvittayakumjorn, Piyawat; Guo, Yike, 2019. Integrating semantic knowledge to tackle zero-shot text classification. In: *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL), Minneapolis, Minnesota, USA*, S. 1031–1040. DOI: 10.18653/v1/N19-1108

- Zhang et al. 2021** Zhang, Xinyi; Koyama, Keisuke; Domae, Yukiyasu; Wan, Weiwei; Harada, Kensuke, 2021.  
A Topological Solution of Entanglement for Complex-shaped Parts in Robotic Bin-Picking.  
In: *IEEE 17th International Conference on Automation Science and Engineering (CASE), Lyon, Frankreich*, S. 461–467.  
ISBN: 1665418737
- Zhou & Chellappa 1988** Zhou, Yi-Tong; Chellappa, Rama, 1988.  
Computation of optical flow using a neural network.  
In: *IEEE International Conference on Neural Networks (ICNN), San Diego, CA, USA*, 71–78 vol.2.  
DOI: 10.1109/ICNN.1988.23914
- Zhu et al. 2017** Zhu, Jun-Yan; Park, Taesung; Isola, Phillip; Efros, Alexei A., 2017.  
Unpaired Image-To-Image Translation Using Cycle-Consistent Adversarial Networks.  
In: *IEEE International Conference on Computer Vision (ICCV), Venedig, Italien*, S. 2223–2232.  
DOI: 10.48550/arXiv.1703.10593

# A Anhang

## A.1 Winkel der Rotationsmodelle in Abhängigkeit der Hemisphärenpfadpose

Tabelle A.1: Winkel des Rotationsmodells (a) in Abhängigkeit der Hemisphärenpfadpose

Hemisphärenpfad- pose $n_P$	Winkel $\alpha(n_P)$ / rad	Winkel $\beta(n_P)$ / rad	Winkel $\gamma(n_P)$ / rad
1	0	0	0
2	0	0	$-\frac{\pi}{6}$
3	0	0	$\frac{\pi}{3}$
4	0	0	$\frac{\pi}{6}$
5	0	0	0
6	0	0	$-\frac{\pi}{6}$
7	0	0	$-\frac{\pi}{3}$
8	0	0	$\frac{\pi}{6}$
9	0	$-\frac{\pi}{12}$	0
10	$-\frac{\pi}{12}$	0	$\frac{\pi}{6}$
11	$-\frac{\pi}{12}$	0	0
12	$-\frac{\pi}{12}$	0	$-\frac{\pi}{6}$
13	0	$\frac{\pi}{12}$	0
14	$\frac{\pi}{12}$	0	$\frac{\pi}{6}$
15	$\frac{\pi}{12}$	0	0
16	$\frac{\pi}{12}$	0	$-\frac{\pi}{6}$
17	0	0	0

Tabelle A.2: Winkel des Rotationsmodells (b) in Abhängigkeit der Hemisphärenpfadpose

Hemisphärenpfad- pose $n_P$	Winkel $\alpha(n_P)$ / rad	Winkel $\beta(n_P)$ / rad	Winkel $\gamma(n_P)$ / rad
1	0	0	0
2	0	0	$-\frac{\pi}{6}$
3	0	0	$\frac{\pi}{3}$
4	0	0	$\frac{\pi}{6}$
5	0	0	0
6	0	0	$-\frac{\pi}{6}$
7	0	0	$-\frac{\pi}{3}$
8	0	0	$\frac{\pi}{6}$
9	0	$-\frac{\pi}{12}$	0
10	0	$-\frac{\pi}{12}$	$-\frac{\pi}{6}$
11	0	$-\frac{\pi}{12}$	$-\frac{\pi}{3}$
12	0	$\frac{\pi}{12}$	$\frac{\pi}{6}$
13	0	$\frac{\pi}{12}$	0
14	0	$\frac{\pi}{12}$	$-\frac{\pi}{6}$
15	0	$\frac{\pi}{12}$	$-\frac{\pi}{3}$
16	0	$-\frac{\pi}{12}$	$\frac{\pi}{6}$
17	0	0	0



