

Bildverarbeitung für ein generisches Entladesystem

zur Erlangung des akademischen Grades eines

Doktors der Ingenieurwissenschaften

von der Fakultät für Informatik
des Karlsruher Instituts für Technologie

genehmigte

D i s s e r t a t i o n

von

Benedikt Kaiser

aus Bielefeld

Tag der mündlichen Prüfung:	15. Dezember 2009
Erster Gutachter:	Prof. Dr.-Ing. Heinz Wörn
Zweiter Gutachter:	Prof. Dr.-Ing. Rüdiger Dillmann

Vorwort

Diese Arbeit entstand im Rahmen meiner Tätigkeit als wissenschaftlicher Mitarbeiter am Institut für Prozessrechentechik, Automation und Robotik des Karlsruher Instituts für Technologie.

Ein großer Teil der Arbeit wurde im Rahmen des vom Bundesministerium für Bildung und Forschung (BMBF) geförderten Forschungsprojektes GENESYS (Generisches Entladesystem) durchgeführt.

Mein besonderer Dank gilt Herrn Prof. Dr.-Ing. Heinz Wörn für die Betreuung der vorliegenden Dissertation und die kontinuierliche Unterstützung meiner Tätigkeit.

Für die Übernahme des Korreferates und für die Durchsicht der Arbeit möchte ich Herrn Prof. Dr.-Ing. Rüdiger Dillmann danken.

Ich bedanke mich bei allen Kolleginnen und Kollegen am Institut für das gute Arbeitsklima. Insbesondere möchte ich Ricardo Alejandro Tauro, Simon Notheis, Swei Jen Chen und Björn Hein erwähnen, die mir immer sehr geholfen haben. Ich danke euch für alles !

Den beiden studentischen Hilfskräften Lars Pfozter und Jannik Steinbring danke ich für ihre wertvolle Unterstützung bei vielen Programmieraufgaben.

Schließlich möchte ich mich bei meiner Familie bedanken, die mich immer unterstützt, ermutigt und für mich da ist.

Inhaltsverzeichnis

1	Einleitung	1
1.1	Motivation und Problemstellung	1
1.2	Ziele der Arbeit	2
1.3	Gliederung	4
2	Stand der Technik	5
2.1	Existierende Objektentnahmesysteme	5
2.2	Greifen	12
2.3	Mensch-Roboter-Interaktion	13
3	Konzept und Systemübersicht	19
3.1	Anforderungen an das Bildaufnahmeverfahren	19
3.1.1	Vergleich verschiedener Bildaufnahmeverfahren	19
3.1.2	Fazit für den Einsatz im Entladesystem	21
3.2	Anforderungen an die Bildverarbeitung	22
3.3	Hardwarearchitektur	23
3.3.1	Aufbau der Entladezelle	23
3.3.2	Kalibrierung der optischen Sensoren	23
3.4	Softwarearchitektur	25
3.4.1	Modularisierung	25
3.4.2	Beschreibung der Komponenten	26
3.5	Ablauf	30
4	Realisierung der Bildverarbeitung	33
4.1	Kalibrierung	33
4.1.1	Kalibrierung der externen 2D-Kamera	33
4.1.2	Kalibrierung des Laserscanners	38
4.1.3	Kalibrierung des am Roboter befestigten 3D-Sensors	42
4.2	Containerlokalisierung und Clipping	47
4.2.1	Ansatz zur Bestimmung der Containerposition	47
4.2.2	Experimente zur Containerlokalisierung	48

4.3	Beschreibung des ICP-Verfahrens	51
4.3.1	Das Verfahren	51
4.3.2	Der kd-Tree	53
4.4	Greifpunktbestimmung in Punktwolken	54
4.4.1	Greifflächenmodell	54
4.4.2	Finden der Greifpunkte	54
4.4.3	Ergebnisse	57
4.5	Greifpunktbestimmung in Tiefenbildern	58
4.5.1	Ansatz	58
4.5.2	Das Tiefenbild	59
4.5.3	Kantendetektion und 2D-ICP	61
4.5.4	Anwendung	64
5	Semi-automatisches Entladesystem	69
5.1	Einführung	69
5.2	Bildschirmbasiertes semi-automatisches Entladesystem	71
5.2.1	Eingabe über einen Bildschirm	71
5.2.2	Berechnung des Sichtstrahlschnittpunktes	71
5.2.3	Untersuchung der Genauigkeit der vorgegebenen Positionen	74
5.3	Laserpointerbasiertes semi-automatisches Entladesystem	76
5.3.1	Laserpointer als Eingabegerät	76
5.3.2	Systemüberblick	77
5.3.3	Kalibrierung des Laserpointertools	80
5.3.4	Experimentelle Ergebnisse	82
6	Zusammenfassung und Ausblick	85
6.1	Beiträge der Arbeit	85
6.2	Ausblick	86
	Glossar	89
	Abbildungsverzeichnis	91
	Tabellenverzeichnis	93
	Literaturverzeichnis	95

1 Einleitung

1.1 Motivation und Problemstellung

Industrieroboter sind heute ein unentbehrliches Werkzeug zur Manipulation von Objekten in der automatisierten Produktion und Objekthandhabung. Roboter werden vielfach für Aufgaben eingesetzt, in denen gleiche Bewegungsabläufe sehr oft wiederholt werden müssen und in denen die Positionen der zu manipulierenden Objekte genau bekannt sind.

Durch den zunehmenden Einsatz von Sensoren und fortgeschrittener Sensordatenverarbeitung können auch Aufgaben automatisiert werden, in denen flexibel auf eine sich ändernde Umwelt reagiert werden muss. In realen Anwendungen werden jedoch definierte Randbedingungen vorausgesetzt. So wird davon ausgegangen, dass die zu manipulierenden Teile eine vorher bekannte Form besitzen und das eine definierte Hintergrund- und Beleuchtungssituation herrscht.

Nach dem derzeitigen Stand der Technik existieren jedoch viele Handhabungs-, Kommissionier- und Entladevorgänge die nicht mit Robotern automatisiert werden können, weil die Teile hinsichtlich ihrer Lage, Form und Gewicht nicht exakt bekannt sind oder durch gegenseitige Verklemmungen nicht bewegt werden können.

Diese Art von Aufgabenstellung tritt in vielen Bereichen auf. In der Fertigung werden Werkstücke oft unsortiert in Behältern angeliefert, um dann weiterverarbeitet zu werden. In Warenverteilzentren werden Pakete und Sendungen aus Behältern oder LKWs entnommen und auf Förderbänder gelegt und auf Flughäfen müssen Gepäckstücke aus Flugzeugcontainern entladen und dem Gepäckverteilungssystem übergeben werden.

In der Produktion und der Logistik besteht ein großer Bedarf, diese derzeit noch manuell ausgeführten Tätigkeiten ganz oder teilweise zu automatisieren. Zum einen stellt die manuelle Entladung oftmals einen Flaschenhals in einer ansonsten automatisierten Logistikkette dar, zum anderen besteht der Wunsch nach Humanisierung bestimmter Tätigkeiten. So müssen bei der Entladung von Gepäckcontainern an Flughäfen üblicherweise zwei Arbeiter innerhalb von 3 Minuten ca. 40 Gepäckstücke mit einem Durchschnittsgewicht von 20kg und einem

Maximalgewicht von 50kg auf ein Förderband heben. Würde diese Tätigkeit zumindest teilweise durch Roboter unterstützt, könnten die gesundheitsschädlichen Auswirkungen dieser Arbeit reduziert werden.

Gegenstand dieser Arbeit ist das Entladen von zufällig angeordneten und hinsichtlich ihrer Position, Orientierung, Form und Gewicht nicht exakt bekannten Objekten aus einem Behälter mit Hilfe eines Roboters. Diese Problemstellung wird auch als das „Griff in die Kiste“-Problem (englisch: bin picking) bezeichnet [30, 3, 8, 38, 42, 53] und stellt in seiner allgemeinen Form ein bis heute nicht vollständig gelöstes Problem dar.

Eine große Herausforderung besteht in der Entwicklung geeigneter Verfahren zur Auswertung und Interpretation von Sensordaten, so dass ein Roboter im Sinne der vorgegebenen Entladeaufgabe optimal gesteuert werden kann.

Um das Entladesystem möglichst universell (generisch) halten zu können, sollen die Verfahren ohne große Anpassungen für die verschiedensten Entladeaufgaben verwendet werden können und unter realen Bedingungen in der Industrie oder im Logistikbereich robust funktionieren.

1.2 Ziele der Arbeit

Schwerpunkt dieser Arbeit ist die Realisierung von 2D- und 3D-Bildverarbeitungsverfahren für den Einsatz in einem generischen Entladesystem. Dazu gehören Verfahren zur Kalibrierung der Sensorik, zur Bestimmung geeigneter Greifpunkte und zur Lokalisierung des Behälters. Es wird angenommen, dass die zu greifenden Objekte in ihrer Form, in ihrer Lage und in ihrer Beschaffenheit nicht exakt bekannt sind. Die vorgestellten Verfahren sollen leicht an unterschiedliche Entladeaufgaben angepasst werden können und in natürlichen, „unkooperativen“ Umgebungen robuste Ergebnisse liefern. Darüber hinaus soll ein Ansatz zur semi-automatischen Greifpunktbestimmung realisiert werden, der als Erweiterung des vollautomatischen Entladesystems zum Einsatz kommen kann.

In der Arbeit sollen Beiträge zu folgenden Teilzielen erbracht werden:

- Entwicklung eines Gesamtkonzeptes und einer Systemarchitektur für ein Entladesystem: Zunächst muss eine geeignete Hard- und Softwarearchitektur für das Entladesystem definiert werden. Die Architektur soll sich durch eine starke Modularität auszeichnen, so dass einzelne Komponenten für verschiedene Entladeaufgaben wiederverwendet werden können, wodurch die Flexibilität des Entladesystems erhöht wird.

- Entwicklung von Kalibrierungsverfahren für die eingesetzten Sensoren: Hier geht es um die sogenannte Hand-Auge-Kalibrierung. Sie ist ein notwendiger Schritt, um sämtliche Sensordaten in ein einziges Koordinatensystem integrieren zu können. Während für die Kalibrierung von 2D-Kameras auf Standardverfahren zurückgegriffen werden kann, müssen für die 3D-Sensorik neue Verfahren entwickelt werden.
- Entwicklung eines Verfahrens zur Containerlokalisierung: Bevor Greifpunkte bestimmt werden können, muss der Behälter erkannt und lokalisiert werden, um diesen vom eigentlichen Inhalt abgrenzen zu können. Insbesondere in realistischen Szenarien kann nicht davon ausgegangen werden, dass sich der Behälter immer exakt an der gleichen Position befinden.
- Entwicklung von 2D- und 3D-Bildverarbeitungsverfahren zur Greifpunktbestimmung: In einem zuvor aufgenommenem Umweltmodell müssen geeignete Greifpunkte bestimmt werden, die dann als Zielpositionen an die Robotersteuerung übergeben werden. Die Greifpunktbestimmung soll ohne exakte Kenntnis der Objektform und -beschaffenheit auskommen. So sollen auch unbekannte oder nur schwer modellierbare Objekte robust entladen werden können. Je nach eingesetzten Sensoren sollen die Algorithmen auf 3D-Daten oder auf Tiefenbildern arbeiten.
- Konzeption und Implementierung einer intuitiven Mensch-Roboter-Schnittstelle zur semi-automatischen Greifpunktbestimmung: Zusätzlich zum voll-automatischen Modus, soll ein semi-automatischer Modus entwickelt werden. In realistischen Szenarien kann die Möglichkeit nicht ausgeschlossen werden, dass das vollautomatische System nicht immer das gewünschte Ergebnis liefert. Wir behandeln dieses Problem mit Hilfe eines interaktiven Modus, in dem ein Mensch das Entladesystem durch die manuelle Vorgabe von Greifpunkten unterstützen kann. Der Wechsel des Betriebsmodus erlaubt es, sich die überlegenen Fähigkeiten des Menschen in der Szenenanalyse zunutze zu machen. Ein solches semi-automatisches System ist nicht nur als eigenständiges System denkbar, sondern es ist auch hilfreich, um neue Entladestrategien manuell zu erproben, ohne komplexen Programmcode schreiben zu müssen. So kann ein Operator testen, wie sich Reihenfolge und Positionen der Greifpositionen auf den Entladevorgang auswirken. Darüber hinaus kann ein interaktiver Modus auch als Benchmark für ein vollautomatisches System dienen, denn die Greifpunkte die durch ein vollautomatisches System ausgewählt werden, müssen sich mit manuell vorgegebenen Greifpunkten hinsichtlich der Gesamtentladezeit vergleichen lassen

können.

Die vorliegende Arbeit entstand im Rahmen des Forschungsprojektes GENE-SYS (Generisches Entladesystem, [1]). Ziel des Projektes war die Entwicklung eines flexiblen, sensor- und roboterbasierten Entladesystems, welches in der Lage ist, Teile handzuhaben, welche in Position und Orientierung, sowie in Form und Gewicht nicht exakt bekannt sind und zusätzlich verklemmt sein können. Das zu entwickelnde System sollte somit einen Beitrag zum „Griff in die Kiste“-Problem liefern und insbesondere im Hinblick auf eine Anwendung im industriellen und Logistikumfeld entwickelt werden.

1.3 Gliederung

Der Arbeit liegt folgende Gliederung zu Grunde:

Kapitel 2 beschäftigt sich mit dem aktuellen Stand der Technik auf dem Gebiet des „Griff in die Kiste“-Problems unter besonderer Berücksichtigung der Sensordatenverarbeitung. Des weiteren wird auf das Thema Greifen eingegangen und das Thema Mensch-Roboter-Interaktion behandelt.

Anschließend wird in **Kapitel 3** das in dieser Arbeit entwickelte Konzept für die Bildverarbeitung in einem generischen Entladesystem vorgestellt. Ebenfalls wird ein Überblick über das Gesamtsystem gegeben und die einzelnen Komponenten des Systems erläutert.

Das **Kapitel 4** hat dann die Realisierung der Bildverarbeitung zum Inhalt. Dazu gehört die Beschreibung verschiedener Kalibrierungsverfahren, der Containerlokalisierung und der Greifpunktbestimmung. Die einzelnen Verfahren werden innerhalb des Gesamtsystems evaluiert und die Ergebnisse dargestellt.

Kapitel 5 beschreibt die Entwicklung und die Evaluierung zweier semi-automatischer Entladesysteme, die mit Hilfe intuitiver Mensch-Roboter-Schnittstellen gesteuert werden.

Die Arbeit endet schließlich in **Kapitel 6** mit einer Zusammenfassung und einem Ausblick.

2 Stand der Technik

Dieses Kapitel behandelt den Stand der Technik für das „Griff in die Kiste“-Problem. Es werden Arbeiten vorgestellt die sensor- und roboterbasierte Entnahmesysteme zum Gegenstand haben. Zusätzlich werden allgemeinere Arbeiten zum Thema Greifen in der Robotik vorgestellt und das Thema Mensch-Roboter-Interaktion anhand verschiedener Beispiele aus der Literatur erläutert.

2.1 Existierende Objektentnahmesysteme

Im Folgenden werden verschiedene sensor- und roboterbasierte Entnahmesysteme anhand der entsprechenden Literatur vorgestellt. Für jedes System wird kurz erläutert welche Arten von Sensoren eingesetzt wurden, wie die Sensordatenverarbeitung prinzipiell realisiert wurde und was für Arten von Objekten mit dem System gehandhabt werden konnten.

Ikeuchi, Horn et al. stellen 1983 ein sensor- und roboterbasiertes System zur Entnahme eines Objektes von einem ungeordneten Objekthaufen vor [31, 30]. Für die Aufnahme der Szene wird das sogenannte „photometrische Stereo“-Verfahren eingesetzt. Dabei wird die Szene von mehreren, weit entfernten Lichtquellen aus unterschiedlichen Richtungen nacheinander beleuchtet und mit einer 2D-Kamera jeweils ein Grauwertbild erzeugt. Aus dem Vergleich der Helligkeitswerte der einzelnen Bildpixel mit den Helligkeitswerten aus einer zuvor berechneten Reflexionskarte, können die Oberflächennormalen in der Szene bestimmt werden. Nach einem Segmentierungsschritt werden Histogramme der Verteilungen der Oberflächennormalen berechnet und mit entsprechenden Histogrammen von zuvor abgespeicherten Modellen verglichen. Daraus lassen sich Position und Orientierung von erkannten Objekte ableiten, die als Greifposition für einen Roboter dienen. Der Roboter kann so an die richtige Position und in die richtige Orientierung verfahren werden, um das Objekt mit Hilfe eines Parallelbackengreifers anzuheben.

In [63] wird ein Ansatz zur Erkennung und Lagebestimmung komplexer Werkstücke vorgestellt, welche anschließend mit einem Roboter gegriffen werden sollen. Mit einem Stereokamerasystem werden zunächst zwei Bilder der Szene aufgenommen. In den Bildern wird nach bestimmten einfachen Objektmerkmalen (z.B. nach

kreisförmigen Öffnungen im Werkstück) gesucht, die eine robuste Identifizierung des Objektes zulassen. Durch Korrespondenzanalyse von Objektmerkmalen im linken und rechten Kamerabild können die Position und Orientierung des Werkstücks bestimmt werden. Das vorgestellte System setzt voraus, dass die Werkstücke zuvor durch einfache Geometrien modelliert worden sind. Die Modellierung wurde im vorgestellten System durch eine Trainingsphase realisiert, welche allerdings durch einen menschlichen Operator überwacht werden muss.

Ein System, welches ausschließlich 2D-Kameras für die Greifpunktbestimmung verwendet, wird in [32] beschrieben. Für das Greifen bekannter, rechteckiger Blechteile wird unter kontrollierten Lichtbedingungen ein Aufnahme der Szene erstellt. Auf dem Grauwertbild wird eine sogenannte Blobanalyse durchgeführt, bei der versucht wird, in einem Bild zusammenhängende, homogene Regionen zu finden. Der Schwerpunkt und die Hauptachsen der größten gefundenen Region geben dann die Position und Orientierung des Greifpunktes vor.

In [40] wird ein roboterbasiertes System zum Depalettieren von Plastiksäcken verwendet. In der beschriebenen Anwendung existiert nur eine Art von Objekten und diese sind lagenweise, nebeneinander angeordnet. Mit Hilfe eines 2D-Kamerabildes wird die Position der Palette detektiert. Da es in der vorgestellten Anwendung zwei Möglichkeiten gibt, wie die Plastiksäcke angeordnet sein können, muss noch die richtige Anordnung bestimmt werden. Ein Portalroboter, der mit einem Vakuumgreifer ausgestattet ist, kann dann die Säcke anheben und der weiteren Verarbeitung zuführen. Der Greifvorgang wird dabei durch einen Näherungssensor überwacht.

Das in [5] vorgestellte System erfasst die Szene mit einer Stereokamera. In dem Tiefenbild wird nach einer ebenen Fläche gesucht, die weit oben liegt und die dafür geeignet ist, dass an dieser ein Vakuumgreifer ansetzen kann. Für den Greifvorgang selbst ist daher kein Vorwissen über die Objekte notwendig. Das gegriffene und damit separierte Objekt kann jedoch in einem nachfolgenden Arbeitsschritt mit einer 2D-Kamera analysiert werden und anhand einer Modelldatenbank identifiziert werden.

Kristensen, Estable et al. stellen in [44] ein „Griff in die Kiste“-System auf Basis einer sogenannten „solid state range“-Kamera vor. Bei dieser Kamera wird in regelmäßigen Abständen ein Laserlichtimpuls ausgesendet. Das reflektierte Signal wird durch einen extrem schnellen Shutter periodisch abgeschnitten. Somit hängt die Menge des vom CCD-Chips registrierten Lichts direkt von der Entfernung zum Objekt ab. Die Kamera liefert so ein Tiefenbild zurück, auf dem eine Ebenensegmentierung durchgeführt wird. Anschließend wird ein modellbasiertes Verfahren zur Erkennung von Quadern verwendet, um geeignete Greifpositionen für einen Vakuumgreifer zu bestimmen. In [44] wird daneben auch noch ein modellfreier

Ansatz vorgestellt, der Greifpositionen für einen Backengreifer finden kann.

Müller und Wörn beschreiben in [53] und [54] ein roboter- und sensorbasiertes System zur Entnahme von ungeordneten und weitgehend unbekanntem Objekten. Das System beruht auf einem 6-Achs Industrieroboter, welcher mit einem Wechselgreifersystem ausgestattet ist. Es können verschiedene Arten von starren oder elastischen Objekten entweder mit einem Parallelbacken- oder einem Vakuumgreifer gegriffen werden. Mit einem Lasertriangulationsverfahren wird ein Tiefenbild der Szene erstellt. Das Tiefenbild wird dann in ebene Flächen segmentiert, die schließlich zur Greifpunktbestimmung verwendet werden. Die Tiefeninformationen werden zusätzlich benutzt, um Szenenbereiche danach zu unterscheiden, ob sie für den Roboter sicher und kollisionsfrei befahrbar sind, ob sie durch Hindernisse belegt sind, oder ob der Belegungszustand unbekannt ist. Mit diesem Wissen soll eine zeitoptimale Greifplanung erreicht werden. In [54] wird das Entnahmesystem durch einen zweiten Roboter erweitert, der sogenannte Untersuchungsoperationen ausführen soll. Eine Untersuchungsoperation ist das versuchsweise Anheben eines Objektes, um festzustellen ob dieses mit einem anderen Objekt oder Objektteil verbunden ist. So soll ungewolltes Mitanheben und versehentliches Mitbewegen anderer Objekte vermieden werden. Das vorgestellte System wurde im Rahmen einer Kommissionieraufgabe bei einem Versandhandel erprobt (siehe Abb. 2.1).

Katsoulas und Bergen stellen in [38] ein Robotersystem zur Depalettierung von Paketen mit bekannten Abmessungen vor. Sie benutzen einen am Roboterflansch befestigten Laserscanner zur Aufnahme eines Tiefenbildes. Darin werden als erstes Kanten und Ecken extrahiert. Aus den Richtungen und den Schnittpunkten der Kanten können mit Hilfe einer Modelldatenbank Objekte erkannt und deren Position und Orientierung bestimmt werden. Auf Basis dieser Informationen kann schließlich der Roboter verfahren werden, der die Objekte mit einem Vakuumgreifer aufnimmt.

In [37] und [39] untersucht Katsoulas eine verbesserte Möglichkeit, um quaderförmige Objekte zu erkennen, um sie dann mit einem Roboter greifen zu können. Wie in [38] werden in einem Tiefenbild Kanten und Ecken extrahiert. Die gefundenen Ecken dienen dazu, sogenannte Superquadriken an günstigen Startpositionen zu platzieren. Superquadriken sind parametrisierbare geometrische Modelle, die eine große Variabilität an ellipsoidenförmigen und quaderförmigen Körpern repräsentieren können. Es existieren robuste Verfahren, um Superquadriken iterativ in 3D-Daten einzupassen, wodurch gleichzeitig die Position und die Orientierung eines Objektes rekonstruiert wird. Der Einsatz von Superquadriken erlaubt es, flexible Objekte zu modellieren und deren elastische Verformungen zu berücksichtigen. Das vorgestellte Verfahren setzt allerdings die ungefähre Kenntnis der Abmessungen der zu findenden Objekte voraus.

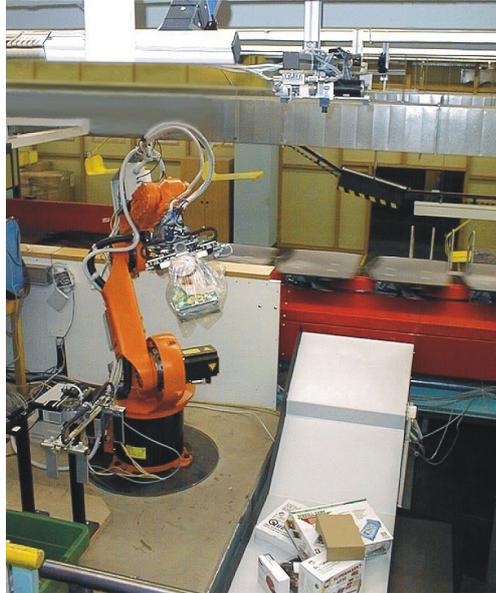


Abbildung 2.1: Das dargestellte Entnahmesystem wurde versuchsweise in einem Versandhandel erprobt. Unterschiedliche, unbekannte Objekte werden dem System über eine Rutsche zugeführt und durch den Roboter gegriffen. (Foto aus [54])

Boughorbel, Zhang et al. stellen in [9] ein Robotersystem mit einem Laserscanner und einer Farbkamera zur Entnahme von bekannten und unbekanntem Objekten aus einem Behälter vor. Eine 3D-Rekonstruktion der Szene wird ebenfalls mit Hilfe von Superquadriken in einzelne Objekte oder Objektteile segmentiert. Die so gewonnenen Positionsdaten werden dann für die Greifpunktbestimmung und die Bahnplanung verwendet. Die 2D-Bilder der Farbkamera werden verwendet, um farbige Marker, die auf dem Roboterarm angebracht worden sind, zu tracken. Dadurch soll die aktuelle Roboterposition überprüft und gegebenenfalls korrigiert werden. Als ein Beispiel für eine reale Anwendung für das vorgestellte Entnahmesystem wird die Sortierung von radioaktivem Abfall genannt.

In [72] wird ein „Griff in die Kiste“-System vorgestellt, welches in der Lage ist, ungeordnete, bekannte Objekte aus einem Behälter zu entnehmen. Dazu wird offline eine Modelldatenbank generiert, die Tiefenbilder eines Objektes in allen möglichen Orientierungen enthält. In einem aufgenommenem Tiefenbild wird an jeder Position für jede Modellansicht ein Korrelationskoeffizient berechnet, welcher die Übereinstimmung zwischen Modell und Daten angibt. Wurden bestimmte Ansichten gefunden, hat man zugleich auch die zugehörigen Greifpositionen- und

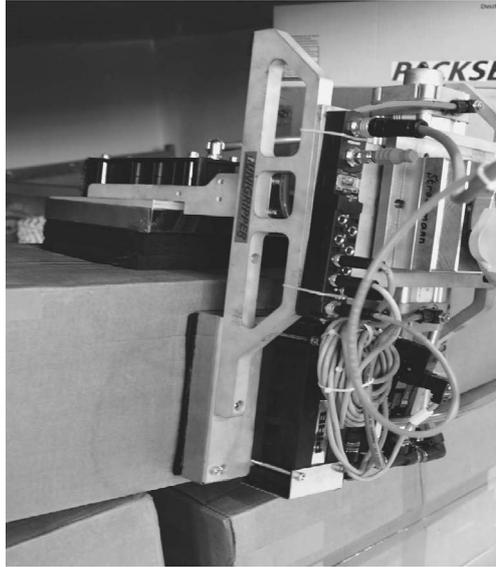


Abbildung 2.2: Prototyp eines Paketentladeroboters. Mit Hilfe eines speziellen Sauggreifers können standardisierte Pakete aus einem LKW entladen werden. (Foto aus [71])

orientierungen. Von allen erkannten Objekten, die kollisionsfrei erreicht werden können, wird das am weitesten oben liegende Objekt gegriffen.

In [20], [79] und [71] wird ein Robotersystem zur Entladung von Paketen aus Containern oder LKWs beschrieben. Das System besteht aus einem schwenkbaren Laserscanner, einem Portalroboter und einem speziell entwickelten Greifer, der kubische Objekte bis 31kg greifen kann (siehe Abb. 2.2). Durch den Einsatz des Portalroboters konnte der Arbeitsbereich und die Taktzeiten gegenüber einem System, welches auf einem Schwenkroboter beruht, verbessert werden. Eine prototypische Anlage wurde in einem Versandzentrum für die Entladung von Paketen aus einem LKW erprobt.

Falls CAD-Modelle der zu greifenden Objekte vorliegen, können diese Daten benutzt werden, um die Objekte zu detektieren und zu lokalisieren. Das Verfahren in [42] wurde für „Griff in die Kiste“-Anwendungen entwickelt und arbeitet auf einer 3D-Punktwolke der Szene. Die Punktwolke wird zuerst durch ein Gittermodell approximiert und in Regionen ähnlicher Oberflächenkrümmung segmentiert. Für jede Region wird anschließend ein sogenannter „Harmonic Shape Context“ (HSC) berechnet, der eine kompakte, lokale Beschreibung einer Oberflächenregion ermöglicht. HSCs können für vorhandene CAD-Daten vorberechnet und in einer Modelldatenbank abgelegt werden. Durch Vergleiche der HSCs werden dann

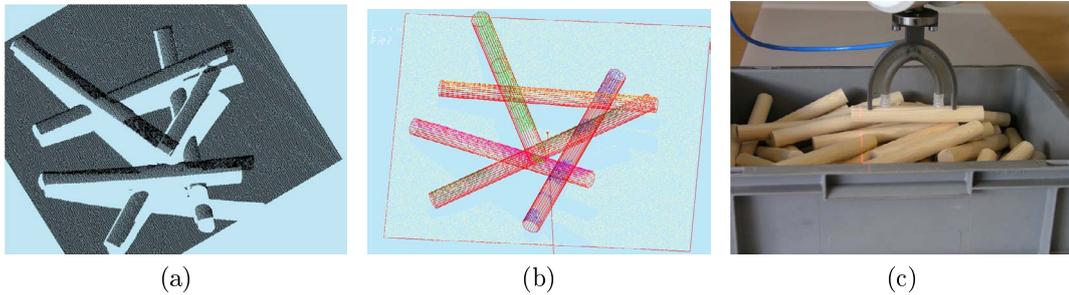


Abbildung 2.3: Die Abbildungen zeigen die aufgenommene Punktwolke (a), eingepasste, einfache Geometrien (Zylinder) (b) und den Greifvorgang unter Einsatz eines speziellen Sauggreifers. (Bilder aus [78])

Position und Orientierung der zu suchenden Objekte bestimmt.

Auch [8] macht sich CAD-Daten für die Lokalisierung bekannter Objekte zu Nutze. Auf Basis der CAD-Modelle werden offline künstliche Tiefenbilder für viele verschiedene Ansichten der Objekte erzeugt. Diese simulierten Ansichten werden mit einem aufgenommenen Tiefenbild verglichen und so eine Liste potentieller Objektpositionen erzeugt. Diese (groben) Objektpositionen dienen als Initialisierung für ein iteratives Verfahren zur Einpassung des CAD-Modells in das Tiefenbild.

Am Fraunhofer IPA wurde ein Verfahren zur Erkennung der Objektlage von Werkstücken für die automatische Handhabung in der Industrie entwickelt [78]. Der Ansatz beruht auf der iterativen Einpassung von einfachen Geometrien (z.B. Zylinder oder Kegel) in eine 3D-Punktwolke. Da viele industrielle Werkstücke gut durch einfache Geometrien (oder Kombinationen einfacher Geometrien) angenähert werden können, soll so eine große Bandbreite (auch komplexer) Teile erkannt werden. Die notwendigen Messdaten werden zuvor durch ein Lasertriangulationsverfahren gewonnen. In einem Demonstrator, bestehend aus einem Roboter und einem speziellen Sauggreifer, konnten erfolgreich zylinderförmige Bauteile aus einem Behälter entnommen werden (siehe Abb. 2.3).

Tabelle 2.1 listet noch einmal einige der beschriebenen Entladesysteme auf. Es wurden nur solche Systeme in der Tabelle aufgeführt, die als funktionsfähiger Demonstrator realisiert worden sind.

Für eine umfassende Auflistung weiterer „Griff in die Kiste“-Systeme aus den Jahren bis einschließlich 2000 sei auch noch auf [55] und [54] verwiesen.

SYSTEM	AUFNHAMETECHNIK	OBJEKTE
Ikeuchi und Horn 1983 [31, 30]	photometrisches Stereo	bekannt, ungeordnet, überlappend (Donuts)
Rahardja und Kosaka 1996 [63]	Stereokamerasystem	bekannt, ungeordnet, überlappend (Industriebauteile)
Ikonen und Kalviainen 1997 [32]	2D-Kamera	flach, rechteckig, bekannt, ungeordnet, überlappend (standardisierte Pakete)
Kavoussanos und Pouliezios 2000 [40]	2D-Kamera	bekannt, lagenweise angeordnet, nicht überlappend (Säcke)
Berger und Bachler 2000 [38]	Stereokamerasystem	unbekannt, ungeordnet, überlappend (industrielle Werkstücke)
Kristensen et al. 2001 [44]	solid state range camera	bekannt, ungeordnet, überlappend (Schachteln)
Müller und Wörn 2001 [54, 53]	Laser-Triangulations-Scanner	unbekannt, ungeordnet, überlappend (Hart- und Weichware)
Katsoulas und Bergen 2002 [38]	am Roboterflansch montierter TOF-Laserscanner	quaderförmig mit bekannten Abmessungen, ungeordnet, überlappend
Boughorbel und Zhang 2003 [9]	Laser-Triangulations-Scanner	unbekannt, ungeordnet, überlappend (geometrisch durch Superquadriken modellierbar)
Schraft und Ledermann 2003 [72]	3D-Laserscanner	bekannt, ungeordnet, überlappend (Metallgehäuse)
Katsoulas 2005 [39]	am Roboterflansch montierter TOF-Laserscanner	quaderförmig mit bekannten Abmessungen, ungeordnet, überlappend (durch Superquadriken modellierbar)
Echelmeyer und Tank 2005 [20, 79]	schwenkbarer TOF-Laserscanner	kubisch, ungeordnet, nicht überlappend (standardisierte Pakete)
Stotz und Verl 2008 [78]	Laser-Triangulations-Scanner	durch einfache Geometrien modellierbare Objekte, ungeordnet, überlappend (industrielle Werkstücke)

Tabelle 2.1: Eigenschaften ausgewählter roboter- und sensorbasierter Entladesysteme. Die aufgelisteten Systeme wurden in Form funktionsfähiger Demonstratoren realisiert.

2.2 Greifen

Das Thema Greifen ist in der Robotik ein aktives Forschungsfeld und für das „Griff in die Kiste“-Problem von zentraler Bedeutung. Im Folgenden sollen allgemeine Aspekte des roboterbasierten Greifens anhand entsprechender Literatur vorgestellt werden. Neben Grundlagen für das Finden stabiler Griffe, werden weitere Ansätze zum Thema Greifen vorgestellt, die zwar das „Griff in die Kiste“-Problem nicht direkt behandeln, aber für die Problemstellung in dieser Arbeit von Interesse sind.

Das Greifen beliebiger, d.h. zuvor nicht bekannter Gegenstände ist ein überaus komplexer Vorgang, bei dem hohe kognitive und motorische Anforderungen bestehen. Das universellste und flexibelste derzeitige Greifsystem ist das des Menschen. Das Zusammenspiel von Objekterkennungsleistungen, taktilen und sensorischen Fähigkeiten und der besondere Aufbau der Hände erlaubt es dem Menschen, die verschiedensten Objekte sicher zu greifen und zu manipulieren.

Im Bereich der Industrierobotik und der Automatisierungstechnik trifft man häufig auf klar definierte Handhabungsaufgaben, bei denen ein bekanntes Werkstück, an vorher festgelegten Positionen gegriffen und abgelegt werden soll. Bei solchen Aufgaben können die Greifpositionen fest einprogrammiert werden. Es kommen speziell an das Werkstück angepasste Greifer zum Einsatz, die zwar wenig flexibel, aber für die jeweilige Aufgabe sehr robust greifen [28, 82].

Die eigentliche Herausforderung beim roboterbasierten Greifen besteht jedoch im sicheren Fassen und Manipulieren von zuvor nicht bekannten Objekten. Die größte Schwierigkeit liegt in der automatischen Auswahl geeigneter Kontaktpunkte an denen ein Greifer ansetzen kann.

Für eine gegebene Anzahl an Kontaktpunkten zwischen einem Objekt und einem Greifer und für eine vorgegebene Objektgeometrie, kann analytisch bestimmt werden, wie die Kontaktpunkte verteilt werden müssen, so dass ein stabiler Griff zustande kommt. Die Kontaktpunkte selbst können dabei mit oder ohne Reibung modelliert werden oder als weiche Kontaktpunkte betrachtet werden. Von der Art der Modellierung hängt es ab, ob bei der Analyse ausschließlich Kräfte entlang der Normalenrichtungen betrachtet werden oder auch solche, die tangential zur Objektoberfläche wirken.

Ein stabiler Griff ist dann erreicht, wenn die Summe aller Kräfte und Momente an den Kontaktpunkten mit allen äußeren Kräften und Momenten, die auf das Objekt wirken im Gleichgewicht sind. Hierbei lässt sich zwischen Kraft- und Formschluss unterscheiden. Bei einem Kraftschluss wird auch die Reibungskraft zwischen Greifer und Objekt berücksichtigt, während bei einem Formschluss das Objekt allein aufgrund der passenden Form gehalten wird.

Für eine Beschreibung verschiedener Verfahren zur Synthese von Greifkontaktpunkten und der Eigenschaften von Griffen siehe z.B. [7], [19] oder [57].

Für die Berechnung von stabilen Griffen bei planaren 2D-Objekten sei auch auf Ponce et al. [61] oder Christopoulos et al. [16] verwiesen. Bei diesen Methoden zur Greifpunktbestimmung werden die Objekte durch ihre 2D-Kontur repräsentiert, welche z.B. aus einem segmentierten 2D-Bild gewonnen werden kann.

In [52] wird ein heuristisches Verfahren zur 3D-Greifplanung für anthropomorphe Roboterhände vorgestellt. Der Ansatz basiert auf der Approximation der zu greifenden Objekte durch einfache geometrische Körper wie z.B. Quader oder Zylinder. Um die große Anzahl möglicher Greiferkonfigurationen berechenbar zu machen, wird zunächst eine kleine Anzahl sinnvoller Greifpositionen heuristisch generiert. Für jede Greiferkonfiguration wird danach die Erreichbarkeit überprüft und ein Qualitätsmaß berechnet [51]. Die Greiferkonfiguration mit der besten Bewertung wird schließlich übernommen.

Die bisherigen Ansätze gehen davon aus, dass ein 2D- oder 3D-Modell des zu greifenden Objektes entweder bereits bekannt ist oder rekonstruiert werden kann [23]. In realistischen Anwendungen ist die korrekte Rekonstruktion von Objekten auf Basis von Sensordaten jedoch oft mit einer großen Unsicherheit behaftet oder aufgrund von unvollständigen Daten nur teilweise möglich.

Saxena et al. [68] schlagen deshalb einen lernenden Algorithmus vor, der ohne 3D-Modell auskommt. In zwei (oder mehr) 2D-Bildern wird unmittelbar nach Positionen gesucht, an denen ein Objekt sinnvoll gegriffen werden kann. Das Verfahren wertet dazu kleine Bildbereiche mit Methoden der Mustererkennung aus und klassifiziert diese nach gültigen und nicht gültigen Greifpositionen. Falls gültige Greifpositionen in verschiedenen 2D-Bildern miteinander korrespondieren, kann die 3D-Position des Greifpunktes durch Triangulation gefunden werden. Das Verfahren hat den Vorteil, dass die Bestimmung von Greifpunkten auch bei unbekanntem Objekten möglich ist.

2.3 Mensch-Roboter-Interaktion

In Kapitel 5 werden neuartige Möglichkeiten zur interaktiven Bedienung eines Entladesystems präsentiert. In diesem Abschnitt geht es deshalb um aktuelle Forschungen im Bereich der Mensch-Roboter-Interaktion und neuartigen Eingabemöglichkeiten für Robotikanwendungen.

Verglichen mit der Erforschung von Mensch-Computer-Schnittstellen [43] ist die Untersuchung von Mensch-Roboter-Schnittstellen (HRI) noch relativ jung. Yanco et al. [84] sowie Scholtz et al. [70] haben verschiedene HRIs zur Steuerung



Abbildung 2.4: Der Mensch zeigt dem Roboterassistenten mit einem Laserpointer an, wo ein Werkstück auf dem Förderband abgelegt werden soll. (Foto aus [77])

von Rettungsrobotern miteinander verglichen und versucht, verschiedene Gütekriterien für die Evaluierung von HRIs anzuwenden. Allgemeine Qualitätsmaße für HRIs werden auch in [75] behandelt.

In [76] und [77] präsentieren Stopp et al. einen mobilen, roboterbasierten Assistenten für den Einsatz in Fabrikationsanlagen. Ziel war es herauszufinden, wie Mensch und Roboter in einer Industrieumgebung effektiv miteinander kooperieren können und wie Robotersysteme intuitiv für neue Aufgaben programmiert werden können. Zu Demonstrationszwecken wurde eine Anwendung realisiert, in der ein menschlicher Operator dem Roboter interaktiv eine Bewegungssequenz vorgeben kann. Der Mensch kann auf einem PDA ein Werkstück auswählen und anschließend mit Hilfe eines Laserpointers auf einem Förderband eine Zielposition anzeigen, an der das Objekt abgelegt werden soll (siehe Abb. 2.4). Weiterhin wurde untersucht, welche Möglichkeiten es gibt, einem mobilen Roboterassistenten per Gesten Befehle zu erteilen.

Mobile Roboterassistentensysteme werden ebenfalls in [26] und [73] beschrieben. Nach einer Definition der Autoren ist Roboterassistenz durch eine direkte Interaktion mit dem System und ein starkes Maß an Flexibilität gekennzeichnet. Es unterstützt den Menschen durch Sensorik, Aktuatorik und Datenverarbeitung, so dass sich die Stärken von Mensch und Roboter sinnvoll ergänzen. Die in [26] und [73] vorgestellten Systeme sollen den Menschen bei manuellen Aufgaben in der Produktion unterstützen, z.B. bei Montage- oder Schweißarbeiten. Gesteuert werden die Systeme über eine Mensch-Maschine-Schnittstelle, welche auch Sprachkommandos entgegennimmt.

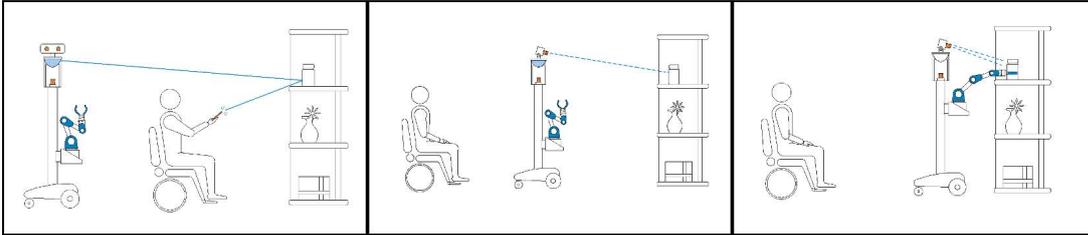


Abbildung 2.5: Mit dem Laserpointer wird ein Gegenstand ausgewählt (links). Die 3D-Position des Laserpunktes wird bestimmt (mitte) und das gewünschte Objekt gegriffen (rechts). (Bilder aus [41])

Eine neuartige eventbasierte Softwarearchitektur zur Interaktion mit Roboterassistenzsystemen wird in [64] vorgestellt. Verschiedene Modalitäten wie Gestenkommandos, Spracheingabe oder Ergebnisse einer Objekterkennung werden miteinander verknüpft und danach in Aktionen transformiert.

In [41] und [15] wird ein mobiles Roboterassistenzsystem zur Unterstützung von körperlich beeinträchtigten Menschen vorgestellt (siehe Abb. 2.5). Der Benutzer hat hierbei verschiedene Möglichkeiten einen Gegenstand aus seiner Umgebung auszuwählen, der dann vom mobilen Robotersystem gegriffen und an einer gewünschten Stelle platziert werden kann. Die Auswahl des Objektes kann über einen Standardlaserpointer oder einen speziellen am Ohr befestigten Laserpointer erfolgen. Der Laserpunkt wird mit einer omnidirektionalen Kamera detektiert und anschließend seine 3D-Position mit Hilfe einer schwenkbaren Stereokamera bestimmt. Als weitere Interaktionsmöglichkeit wird die Vorgabe eines Gegenstandes durch Markieren auf einem Touchscreen genannt. Auch in diesem Fall kann die 3D-Position des Objektes über das Stereokamerasystem ermittelt werden. Auf Basis der 3D-Position wird dann eine Greifposition berechnet, das mobile Robotersystem in die Nähe des Objektes gefahren und ein Greifvorgang gestartet.

Eine Möglichkeit, wie im industriellen Umfeld das sogenannte „Programmieren durch Vormachen“ realisiert werden kann, wird in [49] beschrieben. Zur Programmierung von Bewegungsabläufen, etwa für Schweiß- oder Klebeprozesse, wird der Roboter von Hand entlang der gewünschten Bahn geführt. Die Bahn wird dabei aufgenommen und kann anschließend noch nachbearbeitet werden. Die Interaktion zwischen Robotersystem und Benutzer kann dabei entweder taktil, über eine grafische Schnittstelle oder über Sprachkommandos erfolgen.

Eine innovative Form der Mensch-Roboter-Interaktion wird in [60] vorgestellt. Es wurde untersucht, wie ein Industrieroboter durch aufmalen programmiert werden kann. Mit Hilfe eines digitalen Stiftes, der auf beliebigen Oberflächen funktio-



Abbildung 2.6: Dargestellt ist eine virtuelle Lackierpistole die über kugelförmige Marker getrackt werden kann. Mit Hilfe des Eingabetools können manuell Bahnen generiert werden, die dann von einem Roboter abgefahren werden können. (Foto aus [58])

niert, wird die gewünschte Manipulationsaufgabe aufgezeichnet und automatisch in ein Roboterprogramm umgewandelt. Als Einsatzort für so ein Verfahren werden kleine und mittlere Unternehmen (KMUs) genannt, wo aufgrund kleiner Produktionstückzahlen ein großer Bedarf besteht, Roboter möglichst einfach und flexibel für neue Aufgaben programmieren zu können.

Ein ähnlicher Ansatz zur Mensch-Roboter-Interaktion wird auch in [66] präsentiert. Grundlage ist ein Handheld-Computer auf dem die Karte einer Szene dargestellt ist, in der sich ein mobiler Roboter bewegen kann. Auf dem berührungsempfindlichen Display des Eingabegerätes können mit einem Zeigestift Bahnen aufgezeichnet werden, die dann vom Roboter abgefahren werden. Durch Ausführen bestimmter Zeigegesten auf dem Display, können darüber hinaus vordefinierte Kommandos an den mobilen Roboter übermittelt werden.

Schmid und Hoffmann benutzen ein Touchpad oder eine künstliche Haut, um einen humanoiden Roboter mit einer taktilen Sprache zu kontrollieren [69]. Per Fingerbewegungen kann der Tool Center Point (TCP) des Roboterarms entweder direkt bewegt werden oder es lassen sich komplexe Kommandos (z.B. „Greifen“, „Loslassen“, etc.) über vordefinierte taktile Gesten eingeben.

Nissum et al. stellen in [58] eine virtuelle Umgebung für die Programmierung eines Lackierroboters vor. Der Anwender steht vor einer 3D-Projektionsleinwand, in der ein zu bearbeitendes Werkstück dargestellt wird. Mit Hilfe einer getrackten Lackierpistole (siehe Abb. 2.6) kann ein virtueller Farbauftrag simuliert und der vom Menschen vorgegebene Pfad aufgezeichnet werden. Die Bewegung kann anschließend auf eine Robotersteuerung übertragen und durch den Roboter in der realen Arbeitszelle reproduziert werden.

Ein aktives Forschungsgebiet der Mensch-Roboter-Interaktion ist die roboter-

gestützte Chirurgie [80]. Ein Einsatzgebiet ist die sogenannte minimalinvasive Chirurgie, bei der der Eingriff über kleine Einschnitte mit Hilfe von Endoskopen und speziell konstruierten Instrumenten erfolgt. Durch die Einführung von Teleoperationssystemen [25] kontrolliert der Chirurg die Instrumente nicht mehr direkt, sondern über eine spezielle Nutzerschnittstelle, während die eigentlichen Bewegungen durch einen Roboter ausgeführt werden. Der Chirurg hat dabei volle sechs Freiheitsgrade und wird zusätzlich durch Bewegungsskalierung, Tremorunterdrückung und Stereovision unterstützt. Durch die Integration von haptischem Feedback [48] kann der Grad der Immersivität noch weiter erhöht werden und die Präzision bei der Steuerung des Roboters verbessert werden.

3 Konzept und Systemübersicht

In diesem Kapitel werden zunächst Anforderungen beschrieben, die an Bildaufnahmeverfahren und Bildverarbeitung in einem generischen Entladesystem gestellt werden. Danach wird das Konzept einer komponentenbasierten Hard- und Softwarearchitektur vorgestellt, die leicht an unterschiedliche Entladesysteme und Entladeaufgaben angepasst werden kann.

3.1 Anforderungen an das Bildaufnahmeverfahren

3.1.1 Vergleich verschiedener Bildaufnahmeverfahren

Damit Roboter flexibel auf ihre Umgebung reagieren können, kommen unterschiedliche Verfahren zur Erfassung der Umwelt zum Einsatz: So werden Kräfte und Momente, welche am Endeffektor des Roboters auftreten mit Hilfe von Kraft-/Momentensensorik gemessen, Näherungssensoren dienen zur Gewinnung von Abstandsinformationen, mit Inertialsensoren können Orientierungen bestimmt werden und mit Hilfe von Kameras und Tiefensensoren können 2D- und 3D-Bilddaten erzeugt werden.

Im Rahmen dieser Arbeit wurden verschiedene 3D-Bildaufnahmeverfahren miteinander verglichen und hinsichtlich des Einsatzes in einem Entladesystem beurteilt.

Stereokamerasysteme Bei einem Stereokamerasystem wird ein Objekt oder eine Szene von zwei verschiedenen Kamerapositionen aus aufgenommen. Aufgrund der unterschiedlichen Blickwinkel der Kameras, wird ein und derselbe Punkt auf einem Objekt an verschiedenen Positionen in den Bildebenen der Kameras abgebildet. Der Unterschied der Positionen wird auch als Disparität bezeichnet und ist umso kleiner, je weiter der Punkt von den Kameras entfernt ist. Bei bekannten Kameraparametern und bekannter Anordnung der Kameras, lässt sich aus der Disparität die Entfernung des Objektpunktes bestimmen. Die Herausforderung bei der Berechnung von Tiefeninformationen aus Stereobildern, besteht in der

Bestimmung von korrespondierenden Punkten im linken und rechten Kamerabild (Korrespondenzproblem). Besonders für wenig strukturierte Bereiche stellt dies ein schwieriges Problem dar.

Triangulation mit strukturiertem Licht Bei diesem 3D-Bildaufnahmeverfahren projiziert ein Lichtprojektor spezielle Muster auf die Szene, die dann von einer Kamera aufgenommen werden. Meist wird eine Sequenz von Streifenmustern mit unterschiedlichen Balkenbreiten und Verschiebungen eingesetzt. Bei geeigneter Wahl der Sequenz, kann durch die Folge von hellen und dunklen Kamerapixeln eine eindeutige Korrespondenz zwischen Punkten in der Projektorebene und der Kameraebene hergestellt werden. Ist die Anordnung zwischen Lichtprojektor und Kamera bekannt, kann ähnlich wie bei einem Stereokamerasystem, für jeden Pixel im Kamerabild ein Tiefenwert bestimmt werden.

Laserbasierte Triangulation Bei diesem Verfahren wird in der Szene mit Hilfe von Laserlicht eine Lichtebene erzeugt. Die Lichtebene schneidet sich mit dem zu vermessenden Objekt, so dass auf der Oberfläche des Objektes eine Profillinie entsteht, die von einer Kamera aufgenommen wird. Aus der Projektion der Profillinie im Kamerabild kann dann die Kontur des Objektes berechnet werden. Pro Messung erhält man die Tiefeninformationen jeweils nur für eine Ebene. Für eine flächendeckende Messung muss entweder die Laserebene über die Szene oder das zu vermessende Objekt unter der Laserebene hindurch bewegt werden.

Laserbasiertes Laufzeitverfahren Laserscanner, die auf Basis des Laufzeitverfahrens (Time-of-Flight-Sensor, TOF) arbeiten, messen die Zeit, die ein emittiertes Signal benötigt, um von einem Objekt reflektiert und wieder vom Sensor registriert zu werden. Für die gesuchte Entfernung z ergibt sich dann:

$$z = \frac{ct}{2} \quad (3.1)$$

wobei c die Ausbreitungsgeschwindigkeit des Signals und t die gemessene Zeit ist. In einem Laserscanner befindet sich ein schnell rotierender Spiegel, der den Laser horizontal ablenkt, so dass Entfernungswerte für eine ganze Scanebene zur Verfügung stehen. Um eine flächendeckende Messung zu bekommen, muss die Scanebene relativ zum Objekt bewegt werden.

Photonic Mixer Device Ein Photonic Mixer Device (PMD-Sensor) arbeitet ebenfalls nach dem Laufzeitverfahren. Von einer Infrarotlichtquelle wird ein amplitudenmoduliertes Signal ausgesandt, welches von der Szene reflektiert wird.

Die Laufzeit ergibt sich dann aus der Phasendifferenz zwischen dem aus- und eingehendem Signal. Ein Nachteil bei diesem Verfahren ist die Mehrdeutigkeit bei der Tiefenmessung, da die Phase nur in einem Intervall von $-\pi$ bis $+\pi$ bestimmt werden kann. Mit einem PMD-Sensor kann pro Messung ein vollständiges Tiefenbild aufgenommen werden bei einer Bildwiederholrate von mehr als 10Hz. PMD-Sensoren haben nach dem derzeitigen Stand der Technik eine sehr geringe laterale Auflösung.

3.1.2 Fazit für den Einsatz im Entladesystem

Die Anforderungen an ein Bildaufnahmeverfahren für ein Entladesystem, ergeben sich aus der Art der Entladeaufgabe und den Umgebungsbedingungen am späteren Einsatzort.

In dieser Arbeit wird von einem Einsatz im industriellen Umfeld (z.B. in Fabrikanlagen) oder im Logistikbereich (z.B. in Warenverteilzentren) ausgegangen. Dort herrschen oftmals ungünstige Beleuchtungssituationen, d.h. der Entladebereich ist ungleichmäßig oder schlecht ausgeleuchtet.

Sowohl Stereokamerasysteme als auch Systeme auf Basis strukturierten Lichts liefern unter schwierigen Lichtverhältnissen keine zuverlässigen Ergebnisse. Sensoren auf Basis von Laserlicht sind hier deutlich robuster.

Die triangulationsbasierten Verfahren haben die Eigenschaft, dass ihre absolute Sensitivität für die Tiefenbestimmung mit dem Quadrat der Entfernung abnimmt, während laufzeitbasierte Verfahren über ihrem gesamten Arbeitsbereich einen konstanten Fehler in der Tiefenmessung aufweisen.

Für unsere Anwendung stellte sich ein Laserscanner, der auf einer schwenkbaren Bewegungseinheit montiert ist als praktikable Lösung heraus. Neben einer guten Auflösung kann durch diese Kombination eine relative Unabhängigkeit von der Umgebungsbeleuchtung und der Oberflächenbeschaffenheit der Objekte erzielt werden. Die letzte Eigenschaft ist insbesondere dann wichtig, wenn unbekannte Objekte gehandhabt werden müssen, z.B. bei der Entladung von Stofftaschen oder Koffern aus einem Gepäckcontainer.

Wenn bestimmte Bereiche der Szene, z.B. durch Containerwände, verschattet sind, kann eine fest montierte Laserscanner-Bewegungseinheit die Szene u.U. nicht vollständig erfassen. In solchen Fällen kommt ein am Roboterflansch befestigter PMD-Sensor zum Einsatz. Aufgrund der kleinen Abmessungen des Sensors, kann der Roboter flexibel verfahren, und der Erfassungsbereich des Sensors deutlich erhöht werden. Die geringe Auflösung des PMD-Sensors wird durch Verschmelzung vieler Einzelaufnahmen während einer Roboterbewegung kompensiert.

3.2 Anforderungen an die Bildverarbeitung

Die wesentlichen Aufgaben der Bildverarbeitung für das generische Entladesystem bestehen in der Positionsbestimmung (Lokalisierung) des Behälters und in der Bestimmung geeigneter Greifpositionen für den Roboter.

Positionsbestimmung des Behälters In realistischen Szenarien kann die Position und die Orientierung des Behälters variieren. Eine exakte Lokalisierung des Behälters ist jedoch notwendig, um das Kollisionsmodell (siehe Abschnitt 3.4.2) mit der Realität in Übereinstimmung bringen zu können. Des Weiteren kann der Suchraum für die Greifpunktbestimmung erheblich eingeschränkt werden, wenn bekannt ist, wo sich der Behälter befindet. In der vorliegenden Arbeit wird davon ausgegangen, dass die geometrischen Abmessungen des Behälters bekannt sind, d.h. das Problem der Behälterklassifikation bzw. -identifizierung wird hier nicht behandelt. Ein Verfahren zur Lokalisierung eines Containers wird in Abschnitt 4.2 beschrieben.

Greifpunktbestimmung Eine wichtige Anforderung bei der Entwicklung des generischen Entladesystems besteht darin, dass keine exakte Kenntnis über die Geometrie und die Beschaffenheit der zu greifenden Objekte vorausgesetzt werden soll.

Daraus ergibt sich, dass wir die vielfältigen Methoden zur Objekterkennung [47, 50] für die Greifpunktbestimmung nicht einsetzen können. Bei diesen Verfahren werden in einer Bildaufnahme unbekannte Objekte anhand einer repräsentativen Menge bekannter Objekte klassifiziert und lokalisiert.

Flexiblere Ansätze basieren auf der Einpassung von deformierbaren Modellen in die aufgenommenen Daten. Obwohl diese Verfahren mit einer sehr viel größeren Variabilität an Objekten zurechtkommen, setzen sie dennoch voraus, dass die Objekte durch eine Klasse parametrisierbarer Modelle beschrieben werden können [18, 37].

In der vorliegenden Arbeit wird daher ein einfacherer Ansatz verfolgt, der auf eine Objekterkennung verzichtet. Stattdessen werden Regionen in der Szene bestimmt, an denen ein vorgegebener Greifer ansetzen kann. Dadurch können sinnvolle Greifpunkte für weitgehend unbekannte oder schwer zu modellierende Objekte gefunden werden. Die entwickelten Verfahren werden in den Abschnitten 4.4 und 4.5 erläutert.

3.3 Hardwarearchitektur

3.3.1 Aufbau der Entladezelle

Im Rahmen dieser Arbeit wurden verschiedene Entladezellen aufgebaut. Die Entladezellen unterscheiden sich zwar in einzelnen Hardware- oder Softwarekomponenten, nicht aber in ihrem prinzipiellen Aufbau und der grundsätzlichen Funktionsweise. An dieser Stelle wird deshalb exemplarisch der in Abbildung 3.1 dargestellte Aufbau beschrieben. Je nach Entladeaufgabe werden der verwendete Roboter, die Art des Greifers oder die eingesetzten Sensoren variieren. Die abgebildete Roboterzelle enthält einen 6-Achs Industrieroboter (mit einer Tragfähigkeit von 150kg), der mit einem Vakuumgreifer ausgestattet ist. Vakuumgreifer arbeiten mit Unterdruck und sind für Objekte mit ebenen Oberflächen geeignet. In der Zelle befindet sich ein Container (Abmessungen: 1200mm × 1500mm × 1200mm), in dem sich Pakete befinden. Andere Arten von Objekten sind jedoch ebenfalls möglich. Als optische Sensoren sind eine Farbkamera und ein schwenkbarer Laserscanner vorhanden. Auf einem Standard-PC wird die Sensordatenverarbeitung, die Berechnung der Roboterbewegungen, die Mensch-Maschine-Kommunikation und die Ablaufsteuerung ausgeführt. Die Ausführung der Roboterbewegung und die Ansteuerung der Motoren erfolgt dann in der Robotersteuerung. Eine vereinfachte Darstellung der Hardwarearchitektur ist in Abbildung 3.2 dargestellt.

3.3.2 Kalibrierung der optischen Sensoren

Jeder Sensor gibt Messwerte in Bezug zu einem Sensorkoordinatensystem zurück. Damit ein Roboter anhand von Sensordaten korrekt verfahren werden kann, müssen die relativen Positionen der einzelnen Sensorkoordinatensysteme bezüglich des Roboterkoordinatensystems bekannt sein. Die Bestimmung der notwendigen Transformationen wird auch als Kalibrierung bezeichnet. Je nach Sensorart werden dafür unterschiedliche Verfahren eingesetzt.

Für die Kamerakalibrierung wird ein Kalibrierungsmuster verwendet, welches am Roboterflansch befestigt und mit verschiedenen Roboterpositionen aufgenommen wird. Aus den Roboterpositionen und der Geometrie des Kalibrierungsmusters können die intrinsischen Kameraparameter (Brennweite, optische Verzerrung) und die extrinsischen Kameraparameter (Position und Orientierung) berechnet werden [81, 86].

Für die Kalibrierung des Laserscanners wird eine Kalibrierplatte verwendet, die ebenfalls am Roboterflansch befestigt wird und in verschiedenen Roboterpositionen gescannt wird. Anschließend lässt sich auch für den Laserscanner die gesuchte

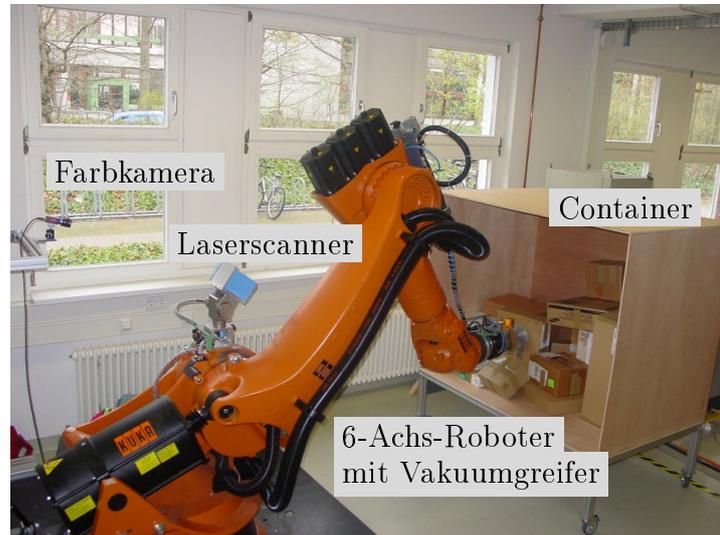


Abbildung 3.1: Die Entladezelle besteht aus einem Industrieroboter mit Vakuumgreifer, verschiedenen Sensoren und einem Container. Hier werden gerade Pakete entladen.

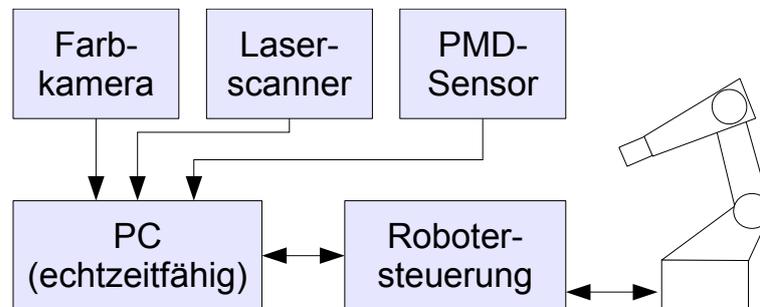


Abbildung 3.2: Vereinfachte Darstellung der Hardwarearchitektur: Die Sensoranbindung und -auswertung erfolgt über einen echtzeitfähigen Standard-PC. Die Roboterbewegungen werden über die Robotersteuerung ausgeführt.

Transformation bestimmen.

Der am Roboterflansch befestigte PMD-Sensor kann über mehrere Aufnahmen einer bekannten und festen Ebene kalibriert werden.

Eine genaue Beschreibung der einzelnen Kalibrierverfahren und die Untersuchung der Genauigkeit der Verfahren erfolgt in Kapitel 4.

3.4 Softwarearchitektur

3.4.1 Modularisierung

Damit das Entladesystem leicht an verschiedene Aufgaben angepasst werden kann, strebten wir bei der Realisierung der Softwarearchitektur eine starke Modularisierung in einzelne Funktionskomponenten an. Die Kommunikation zwischen den Komponenten erfolgt dabei über vorher definierte Schnittstellen. Eine solche Architektur hat mehrere Vorteile:

Wartbarkeit Durch die Aufteilung in kleinere Funktionseinheiten wird die Softwarearchitektur übersichtlicher und Fehler lassen sich leichter identifizieren und korrigieren. Durch die definierten Schnittstellen kann das Ein- und Ausgabeverhalten der einzelnen Softwarekomponenten leichter analysiert werden.

Wiederverwendbarkeit Die Softwarekomponenten können in verschiedenen Konfigurationen miteinander verknüpft oder gegen andere Komponenten ausgetauscht werden, so dass das Gesamtsystem leicht an verschiedenen Entladeaufgaben angepasst werden kann. Dadurch wird eine hohe Wiederverwendbarkeit einzelner Funktionseinheiten erreicht.

Verteilbarkeit Durch die modulare Struktur können die Komponenten bei Bedarf auf verschiedenen Rechnern oder Betriebssystemen ausgeführt werden, um die Ausführungsgeschwindigkeit oder die Verfügbarkeit des Gesamtsystems zu erhöhen.

Im Rahmen dieser Arbeit wurde die „Open Robot Control Software“ (OROCOS) verwendet. Hierbei handelt es sich um eine freie, komponentenbasierte Middleware, die speziell für die Steuerung von Robotern und Maschinen entwickelt wurde [11, 12]. Ein wichtiger Aspekt dieses Softwareframeworks ist die Echtzeitfähigkeit auf Basis eines erweiterten Linux-Betriebssystems.

Abbildung 3.3 zeigt die wesentlichen Komponenten der von uns entwickelten Softwarearchitektur. Die Softwarearchitektur unterteilt sich in einen Echtzeit-

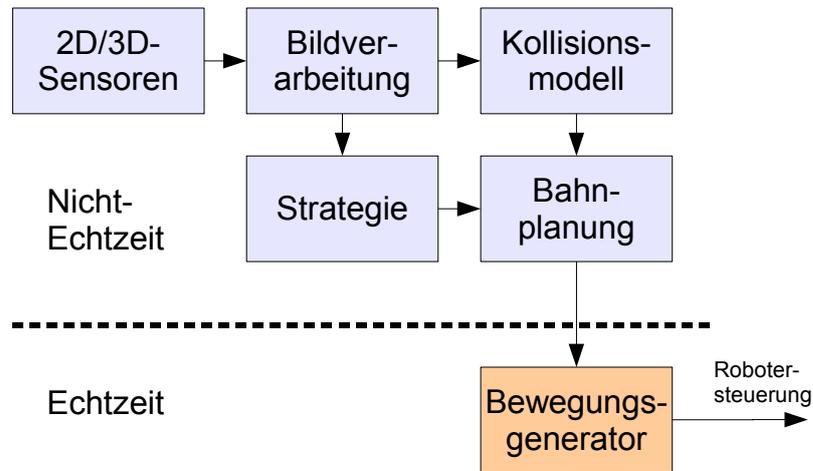


Abbildung 3.3: Vereinfachte Darstellung der Softwarearchitektur: Die Komponenten verteilen sich auf den Echtzeit- und den Nicht-Echtzeitbereich des Betriebssystems. Die Pfeile symbolisieren den Datenfluss zwischen den Komponenten.

und einen Nicht-Echtzeitbereich. Komponenten, die im Echtzeitbereich ausgeführt werden, können über eine Echtzeitschnittstelle direkt mit der Robotersteuerung kommunizieren. Die Einhaltung harter Zeitschranken ist z.B. bei der Ausführung von Roboterbewegungen wichtig, da andernfalls keine glatten Bewegungen garantiert werden können und der Roboter Schaden nehmen könnte. Der Nicht-Echtzeitbereich beinhaltet Komponenten für die Ablaufsteuerung, die Sensordatenverarbeitung, die kollisionsfreie Bahnplanung und die Benutzerschnittstelle.

3.4.2 Beschreibung der Komponenten

Im Folgenden werden die Softwarekomponenten aus Abbildung 3.3 einzeln beschrieben. Nicht dargestellt ist die übergeordnete Ablaufsteuerung, welche die einzelnen Komponenten miteinander synchronisiert und zusätzlich die Benutzerschnittstelle beinhaltet.

2D/3D-Sensoren

Alle eingebundenen Sensoren werden durch je eine Softwarekomponente gekapselt. Die Komponenten besitzen Datenschnittstellen, durch die sie die Sensordaten anderen Komponenten zur Verfügung stellen. Dadurch ist es einfach möglich, eine

Sensorkomponente durch eine andere auszutauschen, falls diese die gleiche Datenschnittstelle implementiert, ohne dass die nachfolgenden Komponenten angepasst werden müssen. Es ist z.B. möglich, dass eine Laserscanner-Komponente, eine PMD-Sensor-Komponente oder eine Stereokamera-Komponente ihre 3D-Daten jeweils über die gleiche Schnittstelle zur Verfügung stellen, so dass die Sensoren flexibel gegeneinander ausgetauscht werden können.

Bildverarbeitung

Die Komponente Bildverarbeitung ist für die Auswertung der 2D- und 3D- Sensordaten zuständig. Die Hauptaufgaben sind die Lokalisierung des zu entladenden Behälters und die Bestimmung potentieller Greifpositionen, an die der Roboter verfahren werden kann. Für die Greifpunktbestimmung wird dabei ein Ansatz verwendet, der ohne exakte Kenntnis der Objektbeschaffenheit oder der Objektgeometrie auskommt. Die Realisierung der Bildverarbeitungskomponente bildet einen Schwerpunkt der vorliegenden Arbeit und wird in Kapitel 4 behandelt.

Strategie

Die Strategiekomponente sortiert und bewertet eine Liste von potentiellen Greifpositionen, die sie zuvor von der Bildverarbeitungskomponente erhalten hat. Zuerst werden alle Greifpositionen verworfen, welche außerhalb des Arbeitsbereiches des Roboters liegen oder für die sich der Roboter in Kollision mit dem Umweltmodell befinden würde. Die verbliebenen Greifpunkte werden anschließend anhand einer Gütefunktion sortiert.

Die Gütefunktion gibt für jeden der Greifpunkte einen skalaren Wert zurück, der ein Maß für die Priorität des Greifpunktes ist. Die Vorschrift für die Berechnung der Gütefunktion ist an die jeweilige Entladeaufgabe angepasst. Die Funktion ergibt sich im Allgemeinen aus einer Heuristik und kann z.B. die Position des Greifpunktes innerhalb des Behälters berücksichtigen.

Ziel der Sortierung ist es, diejenigen Greifpunkte zu priorisieren, die mit einer hohen Wahrscheinlichkeit kollisionsfrei gegriffen und entladen werden können. So soll der gesamte Entladevorgang bezüglich erfolgreicher Greifversuche und kurzer Entladezeit optimiert werden.

Kollisionsmodell

Das Kollisionsmodell stellt die Datenbasis für den Bahnplanungsalgorithmus dar. Bevor der Roboter eine Bahn abfährt, muss diese auf Kollisionen mit Hindernissen

überprüft werden. Dazu enthält das Kollisionsmodell geometrische Beschreibungen des Roboters, der Zellenwände, des Fußbodens, des Behälters und sonstiger Hindernisse. Die Objekte liegen in Form sogenannter CAD-Modelle vor, auf denen sich Kollisionstests effizient berechnen lassen [27, 67].

Der Inhalt des Behälters wird durch eine 3D-Punktewolke, die z.B. durch einen Laserscanner aufgenommen wurde, repräsentiert (siehe Abb. 3.4). Es ist jedoch sehr ineffizient, Kollisionstests mit jedem einzelnen Punkt durchzuführen. Deshalb wird die Punktewolke durch eine Hierarchie von Boundingboxen approximiert (siehe Abb. 3.5). In der ersten Hierarchiestufe wird die Punktewolke durch einen umschließenden Quader angenähert. Falls ein Kollisionstest mit diesem Quader positiv ausfällt, d.h. eine Kollision mit der Punktewolke nicht ausgeschlossen werden kann, wird der Quader in vier gleich große Boundingboxen unterteilt und entsprechende Kollisionstests auf den kleineren Boundingboxen ausgeführt. Dieses Vorgehen wird rekursiv wiederholt, bis eine maximale Rekursionstiefe erreicht wird oder eine Kollision zwischen Roboter und Punktewolke ausgeschlossen werden kann.

Die Approximation der Punktewolke durch Boundingboxen, hat neben der Effizienzsteigerung einen weiteren Vorteil: Da die Boundingboxen bis zur Rückwand des Behälters verlängert werden, ist sichergestellt, dass verschattete, d.h. unbekannte Bereiche als belegt angenommen werden und bei den Kollisionstests als Hindernisse betrachtet werden.

Bahnplaner

Die Bahnplanerkomponente erhält eine priorisierte Liste möglicher Greifpunkte von der Strategiekomponente. In der Reihenfolge der Priorität wird nacheinander versucht, eine kollisionsfreie Bahn zu einer Greifposition zu berechnen. Für diese Aufgabe wird ein effizienter Ansatz auf Basis sogenannter Random-Trees verwendet [45]. Die notwendigen Abstands- und Kollisionsberechnungen werden auf dem zuvor beschriebenen Kollisionsmodell ausgeführt.

Bewegungsgenerator

Der Bewegungsgenerator ist für die Generierung einer abfahrbaren Bewegungstrajektorie und der anschließenden Ausführung der Roboterbewegung verantwortlich. Wenn der Bahnplaner eine kollisionsfreie Bahn zu einer Greifposition gefunden hat, liefert dieser eine Liste von Roboterpositionen zurück. Diese Positionen stellen jedoch nur eine minimale Menge von Stützstellen dar, zwischen denen interpoliert werden muss. Nur so kann die Bahn tatsächlich abgefahren werden. Bei

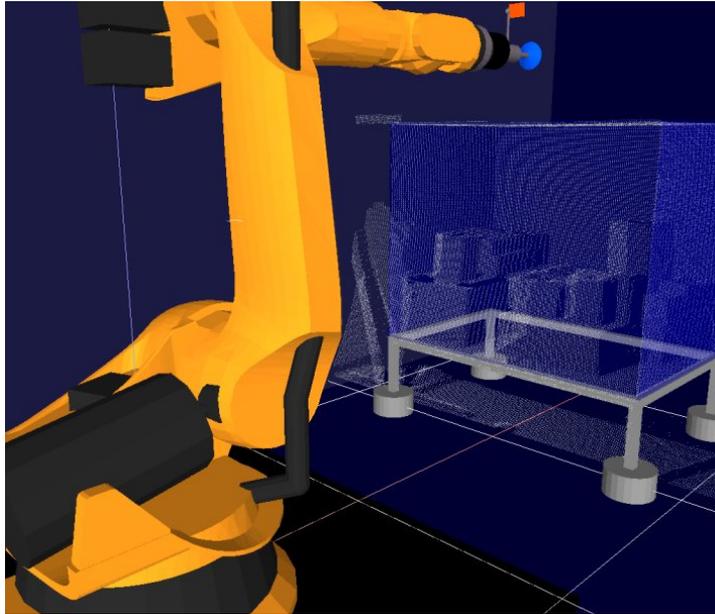


Abbildung 3.4: Um mögliche Kollisionen zwischen Roboter und Hindernissen überprüfen zu können, wird die Entladezelle als CAD-Modell nachgebildet. Zusätzlich ist hier die aufgenommene 3D-Punktwolke überlagert.

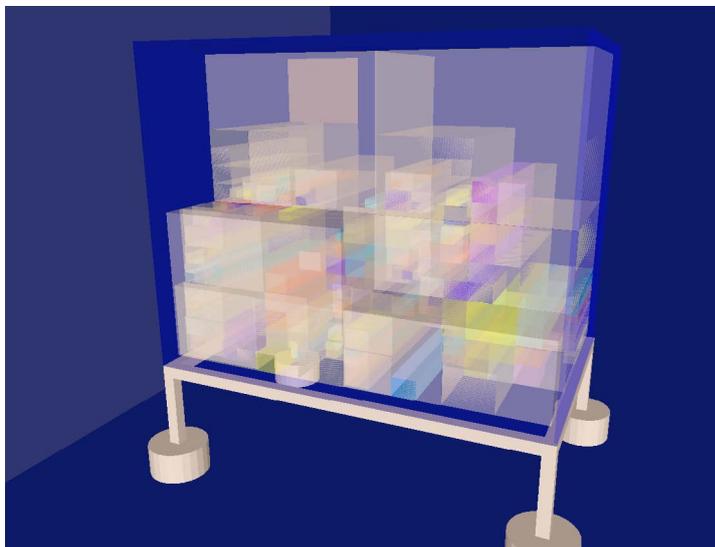


Abbildung 3.5: Die aufgenommene 3D-Punktwolke wird durch eine Hierarchie von Boundingboxen approximiert. So lassen sich Kollisionstests effizient berechnen.

der Ausführung der Bahn muss sichergestellt werden, dass das dazugehörige Geschwindigkeitsprofil keine Unstetigkeiten enthält, so dass vorgegebene maximale Beschleunigungen eingehalten werden können.

Die Komponente ist über eine echtzeitfähige Schnittstelle mit der Robotersteuerung verbunden. Die Robotersteuerung nimmt die aktuellen Positionsdaten entgegen und wandelt sie in Kommandos für die Motorsteuerung um. Der Datenaustausch zwischen der Softwarekomponente und der Robotersteuerung erfolgt im sogenannten IPO-Takt (Interpolationstakt), und liegt im verwendeten System bei 12ms. Um die geforderten harten Zeitschranken sicher einhalten zu können, wird diese Komponente im Echtzeitbereich des Betriebssystem ausgeführt.

3.5 Ablauf

In Abbildung 3.6 ist der Gesamtablauf des automatischen Entladesystems dargestellt. Zunächst wird die Szene dreidimensional erfasst. Aus den 3D-Daten werden die Boundingboxen für das Kollisionsmodell generiert und eine Liste möglicher Greifpunkte erstellt. Erst wenn eine kollisionsfreie Bahn zu einem Greifpunkt gefunden werden konnte wird ein Greifvorgang gestartet. Nachdem das gegriffene Objekt abgelegt wurde, beginnt der nächste Entnahmezyklus mit einer neuen 3D-Aufnahme. Abbildung 3.7 zeigt die erfolgreiche Entnahme eines Objektes.

Es existieren Situationen, in denen kein gültiger Greifpunkt gefunden wird. Im einfachsten Fall ist der Behälter bereits leer, was jedoch leicht anhand der 3D-Daten detektiert werden kann. Unter ungünstigen Bedingungen kann die Greifpunktbestimmung fehlschlagen oder es kann keine kollisionsfreie Bahn gefunden werden. In diesen Fällen wird das Entladesystem in einen semi-automatischen bzw. interaktiven Modus überführt. In diesem Modus kann ein menschlicher Operator Greifpunkte vorgeben und so das automatische System unterstützen. Der semi-automatische Modus wird in Kapitel 5 beschrieben.

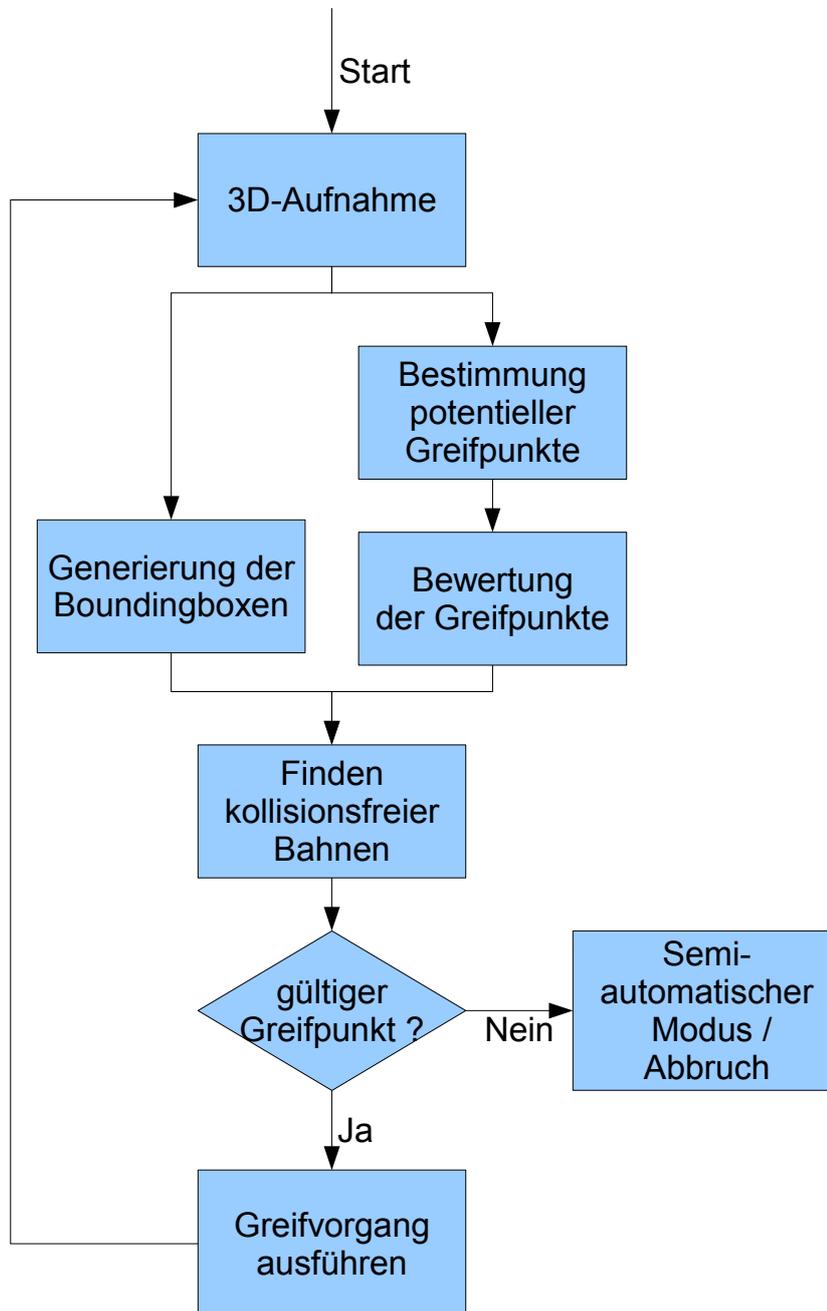


Abbildung 3.6: Ablaufdiagramm des automatischen Entladesystems.



Abbildung 3.7: Entladesequenz (Bilder aus Videoaufnahme). Der Roboter bewegt sich kollisionsfrei zu der berechneten Greifposition (oben) und aktiviert den Vakuumgreifer (mitte). Anschließend verfährt der Roboter mit gegriffenem Objekt zu einer Abladeposition (unten).

4 Realisierung der Bildverarbeitung

Dieses Kapitel behandelt die Realisierung der Bildverarbeitung. Zuerst wird die Kalibrierung der verwendeten Sensoren beschrieben. Anschließend werden die entwickelten Verfahren zur Lokalisierung des Behälters und der Bestimmung geeigneter Greifpositionen erläutert.

4.1 Kalibrierung

Wie in Abschnitt 3.3.2 motiviert wurde, ist eine Kalibrierung der verwendeten Sensoren eine notwendige Vorarbeit für die Weiterverarbeitung der Daten. Ziel ist es, die Sensordaten in ein gemeinsames Weltkoordinatensystem zu überführen, so dass ein Roboter auf Basis der Daten korrekt verfahren werden kann.

In der Robotik existieren unterschiedliche Konzepte zur Darstellung von Koordinatensystemen bzw. Koordinatentransformationen. Im Folgenden werden Positionen als Translationsvektoren und Orientierungen entweder als Rotationsmatrizen oder als Eulerwinkel dargestellt. Ein Koordinatensystem wird durch ein Paar von Position und Orientierung beschrieben. Ein solches Paar wird in der Robotik auch als Frame bezeichnet. Für eine ausführliche Einführung in die mathematische Beschreibung von Koordinatensystemen und Transformationen zwischen Koordinatensystemen sei an dieser Stelle auf [17] verwiesen.

4.1.1 Kalibrierung der externen 2D-Kamera

Bei der Kalibrierung einer 2D-Kamera wird zwischen der intrinsischen und der extrinsischen Kalibrierung unterschieden.

Bei der intrinsischen Kalibrierung werden verschiedene interne Parameter der Kamera und der Optik bestimmt. Der wichtigste Wert ist die Brennweite, welche als Abstand zwischen Objektiv- und Aufnahmeebene definiert ist. Außerdem werden der Bildhauptpunkt (principal point) als Schnittpunkt der optischen Achse

mit der Aufnahmeebene ermittelt, sowie Parameter, welche die optische Verzerrung der Kameraoptik kennzeichnen. Die Kenntnis dieser Werte ermöglicht die geometrische Entzerrung der aufgenommenen Bilder, so dass die Bilder anschließend denen einer idealen Kamera (ohne Verzerrung) entsprechen. Der Brennweitenwert erlaubt die Umrechnung von Pixelabständen in tatsächliche Abstände in der Szene.

Die extrinsischen Parameter definieren die Transformation zwischen dem Kamerakoordinatensystem und einem Referenzkoordinatensystem. Extrinsische Parameter sind also immer von einem anderen Koordinatensystem abhängig.

Kameramodell

Bei der Modellierung der Kamera gehen wir vom sogenannten Lochkameramodell aus. Sei $\vec{p}_c = (x_c, y_c, z_c)^T$ ein 3D-Punkt in Kamerakoordinaten, dann gilt für die normalisierte Projektion (x, y) auf eine Bildebene, welche sich im Einheitsabstand von der Öffnung der Lochkamera befindet:

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} x_c/z_c \\ y_c/z_c \end{pmatrix} \quad (4.1)$$

Dies entspricht einer idealen Abbildung. In der Realität haben wir es jedoch immer mit Linsenverzerrungen zu tun, welche ebenfalls modelliert werden müssen [10]. In der Regel wird dazu folgender Ansatz verwendet:

$$x' = x(1 + k_1 r^2 + k_2 r^4) + 2p_1 xy + p_2(r^2 + 2x^2) \quad (4.2)$$

$$y' = y(1 + k_1 r^2 + k_2 r^4) + p_1(r^2 + 2y^2) + 2p_2 xy \quad (4.3)$$

$$\text{mit } r = \sqrt{x^2 + y^2}$$

Die Parameter k_1 und k_2 bestimmen die radiale Verzerrung, während p_1 und p_2 die tangentielle Verzerrung angeben. Höhere Koeffizienten werden in diesem Modell nicht berücksichtigt, da für die meisten praktischen Anwendungen die so erreichte Genauigkeit ausreichend ist. Die endgültige Projektion (u, v) des Punktes \vec{p}_c auf die Bildebene erhalten wir durch

$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} f_x x' + c_x \\ f_y y' + c_y \end{pmatrix} \quad (4.4)$$

Dabei bezeichnet (c_x, c_y) den Bildhauptpunkt und (f_x, f_y) die Brennweite des optischen Systems in Pixeleinheiten. Falls der Bildaufnahmechip der Kamera aus nichtquadratischen Pixeln besteht, haben f_x und f_y unterschiedliche Werte.

Entzerrung der Bilder

Für die weitere Verarbeitung der Bilder ist es sinnvoll, die Linsenverzerrung der aufgenommenen Bilder auszugleichen. Dies kann effizient implementiert werden, indem für jeden Pixel im entzerrten Bild entsprechend der Gleichungen 4.2, 4.3, und 4.4 die Pixelposition im verzerrten Bild in einer Lookup-Tabelle vorberechnet werden.

Kamerakalibrierung nach Zhang

Die Kalibrierung der Kameraparameter erfolgt nach der Methode von Zhang [86]. Dazu wird eine planare Kalibrierplatte (Schachbrettmuster) unter verschiedenen Orientierungen und Positionen aufgenommen und in jeder Aufnahme die 2D-Positionen bestimmter Merkmalspunkte bestimmt. Leicht zu detektierende Merkmalspunkte sind z.B. die Berührungspunkte von hellen und dunklen Quadraten eines Schachbrettmusters (siehe Abb. 4.1). Jeder Merkmalspunkt hat eine 3D-Koordinate in einem frei wählbaren Koordinatensystem, das fest mit der Kalibrierplatte verknüpft ist. Seien $m_{ij} = (u_{ij}, v_{ij})^T$ die Bildkoordinaten des j -ten Merkmalspunktes in der i -ten Aufnahme, dann lässt sich folgende Fehlerfunktion aufstellen:

$$E = \sum_{i=1}^N \sum_{j=1}^M \|m_{ij} - \tilde{m}_{ij}(A, R_i, t_i, M_j)\| \quad (4.5)$$

$$\text{mit } A = (f_x, f_y, c_x, c_y, k_1, k_2, p_1, p_2)$$

wobei N die Anzahl der Aufnahmen und M die Anzahl der Merkmalspunkte im Kalibrierungsmuster ist. \tilde{m}_{ij} bezeichnet die geschätzte Projektion des Merkmalspunktes M_j in Abhängigkeit der Modellparameter A und der i -ten Transformation $T_{KM_i} = [R_i | t_i]$ zwischen dem Kamerakoordinatensystem und dem Koordinatensystem des Kalibrierungsmusters. Die Rotationsmatrix R_i und der Translationsvektor t_i werden auch als die extrinsischen Modellparameter bezeichnet. Die Fehlerfunktion gibt also die Summe der Abstände zwischen tatsächlichen Merkmalspositionen und den geschätzten Merkmalspositionen an. Bei der Methode nach Zhang werden zunächst die intrinsischen Parameter ohne die Verzerrungsparameter, sowie die extrinsischen Parameter durch eine geschlossene Lösung bestimmt. Anschließend wird die Fehlerfunktion 4.5 einschließlich der Verzerrungsparameter durch ein nichtlineares Optimierungsverfahren minimiert. Für eine detaillierte Beschreibung des Verfahrens sei auf [86] verwiesen.

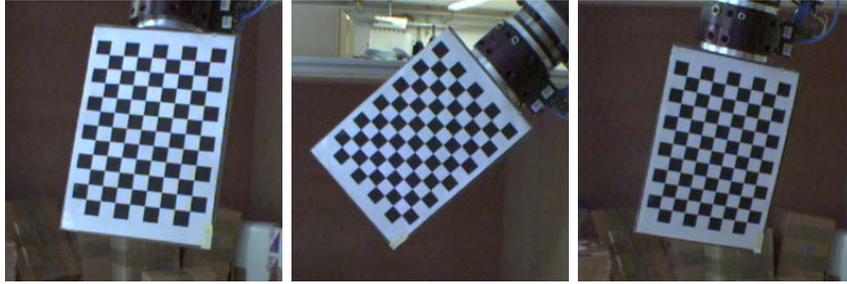


Abbildung 4.1: Das am Roboterflansch befestigte Kalibriermuster wird in verschiedenen Orientierungen und Positionen aufgenommen und die Berührungspunkte der Quadrate als Merkmalspunkte detektiert.

Hand-Auge-Kalibrierung nach Tsai und Lenz

Damit die Bilddaten der Kamera für die Steuerung des Roboters genutzt werden können, muss die Transformation T_{WK} zwischen dem Weltkoordinatensystem und dem Kamerakoordinatensystem bekannt sein (siehe Abb. 4.2). Die gesuchte Transformation wird mit Hilfe des Verfahrens von Tsai und Lenz [81] bestimmt. Üblicherweise wird dieses Verfahren eingesetzt, um die Position und Orientierung einer am Roboterflansch (Hand) montierten Kamera (Auge) bezüglich des Roboterflansches zu bestimmen. Dazu führt der Roboter mitsamt der Kamera eine Folge von Bewegungen aus und nimmt nach jeder Bewegung ein Bild von einem festen Kalibriermuster auf. Anschließend werden in jedem Bild die Merkmalspunkte extrahiert, um die Kamerapositionen bezüglich des Kalibriermusters zu bestimmen.

In unserer Anwendung ist die Kamera fest in der Roboterzelle montiert, während das Kalibriermuster am Endeffektor befestigt ist und zusammen mit dem Roboter bewegt wird. Diese Konfiguration unterscheidet sich von der ursprünglichen Hand-Auge-Konfiguration nur durch Vertauschung der Kamera- und der Kalibriermusterpositionen, d.h die beiden Konfigurationen sind äquivalent zueinander.

Während der Kalibrierung der Kameramodellparameter werden die extrinsischen Parameter T_{KM_i} mitgeschätzt. Diese werden auch für die Hand-Auge-Kalibrierung benötigt, so dass die beiden Kalibrierungen in einem Durchgang durchgeführt werden können.

Im Folgenden soll nur der Ansatz des Verfahrens vorgestellt werden. Für eine ausführliche Beschreibung verschiedener Lösungsmöglichkeiten sei z.B. auf [81, 74, 29] verwiesen.

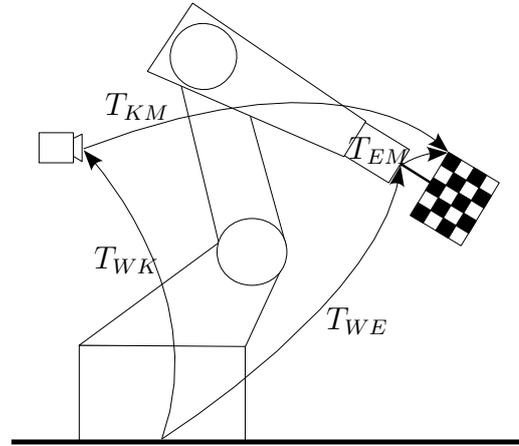


Abbildung 4.2: Für die Kalibrierung der Kameraposition wird eine geschlossene kinematische Kette betrachtet. Diese beinhaltet die Transformationen T_{XY} zwischen den Koordinatensystemen der Kamera (K), des Kalibrieramusters (M), des Endeffektors (E) und der Welt (W).

Seien T_{KM1} und T_{KM2} die Transformationen zwischen Kamera und Kalibriermuster für die Roboterposition 1 bzw. 2 und seien T_{WE1} bzw. T_{WE2} die entsprechenden Transformationen zwischen Welt und Endeffektor. Dann gilt

$$T_{EM} = T_{WE1}^{-1} T_{WK} T_{KM1} = T_{WE2}^{-1} T_{WK} T_{KM2} \quad (4.6)$$

Dies lässt sich umformen zu

$$AX = XB \quad (4.7)$$

mit

$$\begin{aligned} A &= \begin{pmatrix} R_A & \vec{t}_A \\ 0 & 1 \end{pmatrix} = T_{WE2} T_{WE1}^{-1} \\ B &= \begin{pmatrix} R_B & \vec{t}_B \\ 0 & 1 \end{pmatrix} = T_{KM2} T_{KM1}^{-1} \\ X &= \begin{pmatrix} R_X & \vec{t}_X \\ 0 & 1 \end{pmatrix} = T_{WK} \end{aligned}$$

Dabei repräsentiert A die Transformation zwischen den zwei Positionen des Endeffektors, B die entsprechende Transformation zwischen den beiden Positionen des Kalibrieramusters und X stellt die gesuchte Welt-Kamera-Transformation dar.

Gleichung 4.7 läßt sich in zwei Matrixgleichungen zerlegen:

$$R_A R_X = R_X R_B \quad (4.8)$$

$$(R_A - I)\vec{t}_X = R_X \vec{t}_B - \vec{t}_A \quad (4.9)$$

Zur eindeutigen Bestimmung von R_X und \vec{t}_X werden mindestens $n \geq 3$ Aufnahmen in unterschiedlichen Positionen benötigt. Es ergibt sich ein Gleichungssystem von $n-1$ Matrixgleichungen:

$$A_{12}X = XB_{12} \quad (4.10)$$

⋮

$$A_{n-1,n}X = XB_{n-1,n} \quad (4.11)$$

wobei $A_{i-1,i}$ und $B_{i-1,i}$ analog zu Gleichung 4.7 für die Roboterpositionen $i-1$ und i definiert sind. Zur Lösung dieses Gleichungssystems existieren verschiedene geschlossene Lösungen aus der linearen Algebra.

4.1.2 Kalibrierung des Laserscanners

Für die Erfassung einer 3D-Punktwolke wird ein zeilenbasierter Laserscanner verwendet. Dieser ist auf einer Neigeeinheit montiert, um ein flächendeckendes Tiefenbild zu erhalten. Die Transformation der Messwerte in das Weltkoordinatensystem setzt sich daher aus zwei Teilen zusammen: Der Transformation T_{WN} zwischen dem Weltkoordinatensystem und dem Neigeeinheitkoordinatensystem und der Transformation $T_{NS}(\alpha)$ zwischen dem Neigeeinheitkoordinatensystem und dem Sensorkoordinatensystem (siehe Abb. 4.3). Die letztere Transformation ist vom aktuellen Neigewinkel α abhängig.

Die Transformation T_{NS} wurde in unserem System nicht automatisch kalibriert, da die Abmessungen der Neigeeinheit und des Laserscanners, sowie die Position des Sensorursprungs durch die technischen Spezifikationen des Herstellers bekannt sind.

Setzt man das Neigeeinheitkoordinatensystem in den Drehpunkt des Neigeelenkes und geht von einer Drehung um die y -Achse aus, so ergibt sich

$$T_{NS}(\alpha) = (R_{NS}|\vec{t}_{NS}) \quad (4.12)$$

mit

$$R_{NS} = \begin{pmatrix} \cos(\alpha) & 0 & \sin(\alpha) \\ 0 & 1 & 0 \\ -\sin(\alpha) & 0 & \cos(\alpha) \end{pmatrix}$$

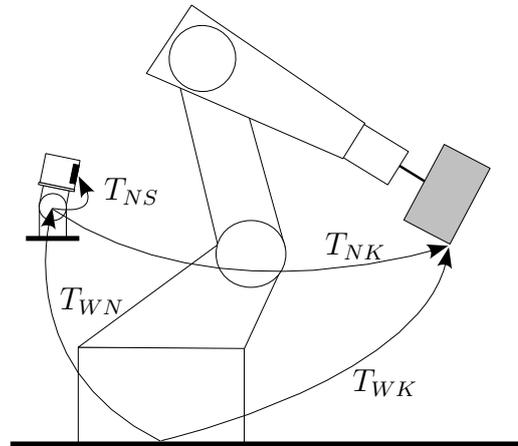


Abbildung 4.3: Für die Kalibrierung des Laserscanners werden die Transformationen zwischen Sensor (S), Neigeeinheit (N), Welt (W) und Kalibrierplatte (K) betrachtet.

Der Translationsvektor \vec{t}_{NS} ergibt sich dabei aus den spezifizierten Abmessungen. Mit Hilfe der Transformation T_{NS} können gescannte Punkte, welche unter verschiedenen Neigewinkeln gemessen wurden, in einem gemeinsamen Koordinatensystem (nämlich dem der Neigeeinheit) zu einer 3D-Punktwolke vereint werden.

Zur Bestimmung der Transformationsmatrix T_{WN} kommt ein Kalibrierkörper zum Einsatz, der am Roboterflansch befestigt wird. Diese Kalibrierplatte ist eine rechteckige Platte mit bekannten Abmessungen (siehe Abb. 4.4). Das Robotersystem bietet herstellereitig die Möglichkeit, am Roboter befestigte Werkzeuge einzumessen, d.h. die Transformation zwischen Werkzeugkoordinatensystem und Flanschkoordinatensystem bestimmen zu lassen. Diese Einmessung wird auch für die Kalibrierplatte durchgeführt, so dass anschließend für jede Roboterposition auch die Position und Orientierung T_{WK} der Kalibrierplatte bezüglich des Weltkoordinatensystems bekannt sind.

Für die eigentliche Kalibrierung wird eine 3D-Punktwolke der Kalibrierplatte durch den schwenkbaren Laserscanner aufgenommen (siehe Abb. 4.5). In der Punktwolke wird die Position und die Orientierung der Platte mit Hilfe eines modellbasierten, iterativen Einpassungsverfahrens bestimmt (siehe Abschnitt 4.3). Das Ergebnis ist die Transformation T_{NK} zwischen dem Neigeeinheitkoordinatensystem (in dem die Punktwolke dargestellt ist) und der Kalibrierplatte.

Die gesuchte Transformation zwischen dem Weltkoordinatensystem und der

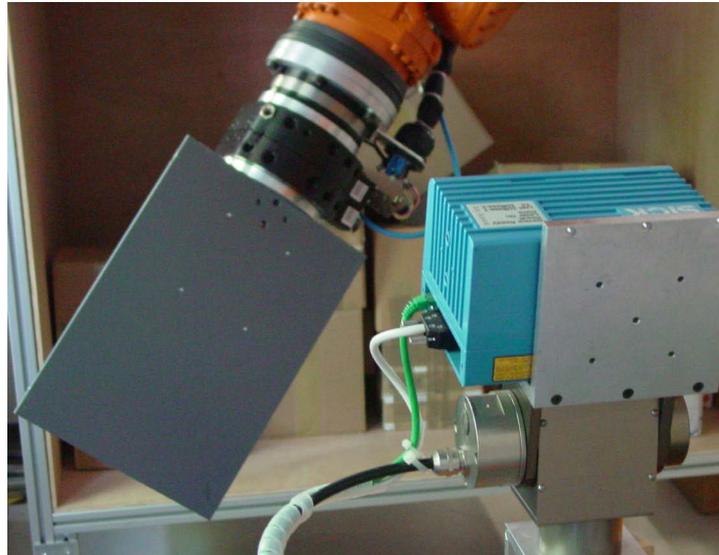


Abbildung 4.4: Die Kalibrierplatte und der Laserscanner, welcher auf einer Neigeeinheit montiert ist.

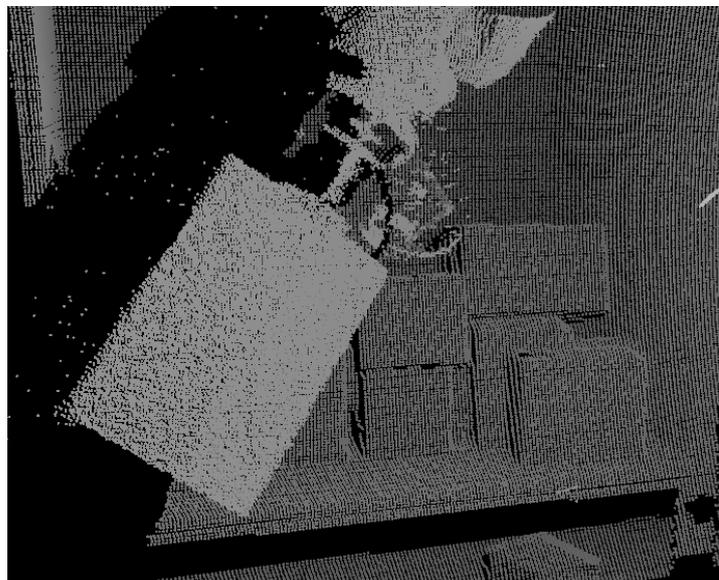


Abbildung 4.5: Mit dem Laserscanner aufgenommene Punktwolke der Kalibrierplatte

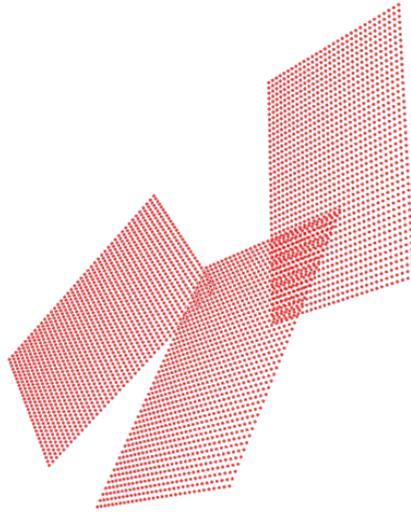


Abbildung 4.6: Für die Laserscannerkalibrierung wird das Modell eines virtuellen Kalibrierkörpers eingepasst. Dieser ergibt sich durch Überlagerung einzelner Kalibrierplatten.

Neigeinheit ergibt sich dann zu

$$T_{WN} = T_{WK}T_{NK}^{-1} \quad (4.13)$$

Um die Robustheit des Verfahrens zu erhöhen, wird die Platte in verschiedenen Positionen und Orientierungen aufgenommen und die einzelnen Punktwolken zu einer einzigen, großen Punktwolke zusammengefasst. Dadurch erhält man einen „virtuellen“ Kalibrierkörper, der aus mehreren, nicht koplanaren Flächen besteht. Statt die Transformation einer Fläche, wird entsprechend die Transformation des virtuellen Kalibrierkörpers bestimmt (siehe Abb. 4.6).

Experimentelle Ergebnisse

In diesem Abschnitt werden experimentelle Ergebnisse der Laserscannerkalibrierung vorgestellt.

In Tabelle 4.1 sind die Ergebnisse für insgesamt zehn verschiedene Kalibrierungen der Laserscanner-Transformation zu sehen. Die Transformationsmatrix ist durch die Parameter x , y , z für die Translation und die drei Euler-Winkel α , β und γ für die Rotation beschrieben. Für jede Kalibrierungen wurde die Kalibrierplatte in eine neue Positionen bewegt und jeweils nur eine Punktwolke aufgenommen.

	x [mm]	y [mm]	z [mm]	α [°]	β [°]	γ [°]
Kalib. 1	324,63	83,88	1200,54	1,49	0,81	-0,14
Kalib. 2	324,68	87,27	1195,82	1,24	0,56	0,14
Kalib. 3	322,96	83,09	1186,57	1,43	0,07	0,46
Kalib. 4	323,54	83,28	1198,97	1,45	0,60	0,33
Kalib. 5	324,07	83,84	1195,91	1,48	0,49	0,22
Kalib. 6	323,17	83,41	1196,31	1,39	0,50	0,34
Kalib. 7	322,65	83,73	1188,12	1,40	0,13	0,18
Kalib. 8	325,56	87,61	1203,71	1,45	1,01	-0,37
Kalib. 9	326,8	82,64	1189,38	1,66	0,30	-0,07
Kalib. 10	324,5	90,61	1202,38	1,14	1,11	-0,25
Std.abw.	1,271	2,632	6,001	0,141	0,345	0,276

Tabelle 4.1: Ergebnisse der Laserscanner-Kalibrierung mit Einfachscan.

Table 4.2 zeigt die Ergebnisse von zehn weiteren Kalibrierungsdurchgängen. Dieses Mal wurden für jede Kalibrierung drei Aufnahmen von unterschiedlich positionierten Kalibrierplatten verwendet.

Um ein Maß für die Stabilität des Kalibrierverfahrens zu bekommen, wurden die Standardabweichungen über die Schätzwerte der einzelnen Parameter berechnet.

Der Vergleich zwischen den Tabellen 4.1 und 4.2 zeigt, dass die Ergebnisse deutlich weniger schwanken, wenn mehrere Aufnahmen für die Kalibrierungen verwendet werden.

4.1.3 Kalibrierung des am Roboter befestigten 3D-Sensors

Ein relativ neuer Typ von 3D-Time-of-Flight-Kameras, das sogenannte Photonic-Mixer-Device (PMD) wird in [83] vorgestellt. In [46] werden Methoden zur intrinsischen Kalibrierung von PMD-Sensoren präsentiert. Im Folgenden wird dagegen die extrinsische Hand-Auge-Kalibrierung des PMD-Sensors behandelt. Das vorgestellte Verfahren macht sich die Eigenschaft zunutze, dass dieser 3D-Sensor bei jeder Aufnahme ein vollständiges Tiefenbild zur Verfügung stellt.

Zunächst wird das Problem der Hand-Auge-Kalibrierung des PMD-Sensors erläutert und eine geschlossene Lösung für dieses Problem gegeben. Anschließend werden zwei Verbesserungsmöglichkeiten für das Kalibrierungsverfahren beschrieben [34].

	x [mm]	y [mm]	z [mm]	α [°]	β [°]	γ [°]
Kalib. 1	324,87	88,82	1195,67	1,25	0,61	-0,14
Kalib. 2	324,6	87,06	1198,59	1,33	0,75	-0,23
Kalib. 3	324,52	89,99	1195,54	1,22	0,60	-0,29
Kalib. 4	324,67	86,69	1197,65	1,30	0,66	0,01
Kalib. 5	324,46	91,42	1193,46	1,18	0,49	-0,07
Kalib. 6	325,32	89,03	1193,70	1,25	0,56	-0,01
Kalib. 7	324,35	87,32	1196,91	1,28	0,60	0,19
Kalib. 8	324,98	86,59	1199,00	1,33	0,74	-0,07
Kalib. 9	324,20	86,59	1195,87	1,30	0,55	0,18
Kalib. 10	324,22	85,38	1197,20	1,38	0,62	0,03
Std.abw.	0,354	1,861	1,871	0,055	0,084	0,155

Tabelle 4.2: Ergebnisse der Laserscanner-Kalibrierung mit Mehrfachscan.

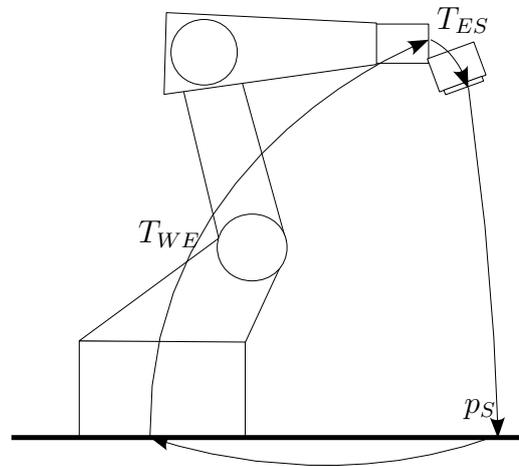


Abbildung 4.7: Für die Hand-Auge-Kalibrierung des PMD-Sensors wird eine kinematische Kette aus Welt (W), Endeffektor (E), Sensor (S) und Fußboden betrachtet.

Extrinsische Sensorkalibrierung

Position und Orientierung des Sensors in Bezug auf ein Referenzkoordinatensystem können durch einen Translationsvektor \vec{t} und eine 3x3 Rotationsmatrix R beschrieben werden. Translation und Rotation können zu einer homogenen Transformationmatrix T kombiniert werden.

Unter Hand-Auge-Kalibrierung versteht man die Bestimmung von Position und Orientierung eines am Roboterflansch befestigten Sensors in Bezug auf den Endeffektor, d.h. die Bestimmung der Transformation T_{ES} vom Endeffektorkoordinatensystem in das Sensorkoordinatensystem (siehe Abb. 4.7).

Ist die Transformation T_{WE} vom Weltkoordinatensystem in das Endeffektorkoordinatensystem bekannt, kann ein Punkt \vec{p}_S , der vom 3D-Sensor aufgenommen wurde, in Weltkoordinaten ausgedrückt werden:

$$\vec{p}_W = T_{WE}T_{ES}\vec{p}_S \quad (4.14)$$

In [85] wird ein Kalibrierungsverfahren beschrieben, welches auf der Messung eines ebenen Kalibrierkörpers durch einen Laserscanner beruht.

In ähnlicher Weise sollen Aufnahmen des ebenen Fußbodens der Roboterzelle benutzt werden, um die Transformation vom Endeffektor- in das Sensorkoordinatensystem zu bestimmen.

Zunächst müssen die Koordinaten von (mindestens) drei Punkten des Fußbodens bestimmt werden. Dies kann z.B. mit einer am Roboter befestigten Messspitze geschehen. Drei Punkte sind notwendig, um den Normalenvektor \vec{n}_W und einen beliebigen weiteren Punkt \vec{p}_W^0 der Ebene zu erhalten.

Während der Kalibrierungsprozedur erstellt der PMD-Sensor mehrere 3D-Aufnahmen des Fußbodens. Der Sensor wird mit dem Roboter für jede Aufnahme in eine neue Position und Orientierung verfahren.

Aus jeder 3D-Aufnahme kann der Normalenvektor $\vec{n}_{S,i}$ (mit $i \in [1..N]$ und N = Anzahl der Aufnahmen) und ein beliebiger Punkt $\vec{p}_{S,i}$ des Fußbodens, jeweils ausgedrückt in Sensorkoordinaten, bestimmt werden. Zusätzlich sind die jeweilige Position $\vec{t}_{WE,i}$ und die Orientierung $R_{WE,i}$ des Endeffektors bekannt.

Dann gilt für die Rotationsmatrix R_{ES} der Hand-Auge-Transformation:

$$R_{ES}\vec{n}_{S,i} = \vec{n}_{E,i} = R_{WE,i}^{-1}\vec{n}_W \quad (4.15)$$

Werden alle Vektoren $\vec{n}_{E,i}$ und $\vec{n}_{S,i}$ der Aufnahmen zu zwei 3xN-Matrizen A und B zusammengefasst, ergibt sich folgendes lineares Gleichungssystem.

$$A = R_{ES}B \quad (4.16)$$

mit

$$A = [\vec{n}_{E,1} \dots \vec{n}_{E,N}] \quad \text{und} \quad B = [\vec{n}_{S,1} \dots \vec{n}_{S,N}]$$

Für den Fall $N > 3$, ist dieses Gleichungssystem überbestimmt. Es handelt sich hierbei um ein lineares Regressionsproblem, für das eine geschlossene Lösung unter Verwendung der sogenannten Pseudoinversen existiert. Wir erhalten:

$$R_{ES} = AB^T[BB^T]^{-1} \quad (4.17)$$

Für die Bestimmung der vollständigen Hand-Auge-Transformation fehlt noch der Translationsvektor \vec{t}_{ES} . Für dessen Bestimmung drücken wir zunächst den Normalenvektor \vec{n}_W der Fußbodenebene und einen beliebigen weiteren Punkt \vec{p}_W^0 der Ebene in Endeffektorkoordinaten aus:

$$\vec{n}_{E,i} = R_{WE,i}^{-1} \vec{n}_W \quad (4.18)$$

$$\vec{p}_{E,i} = T_{WE,i}^{-1} \vec{p}_W^0 \quad (4.19)$$

Wird in jeder 3D-Aufnahme ein beliebiger Punkt $\vec{p}_{S,i}$ der Fußbodenebene bestimmt, dann muss nach der hesseschen Normalenform für Ebenen gelten:

$$\vec{n}_{E,i}(R_{ES} \vec{p}_{S,i} + \vec{t}_{ES} - \vec{p}_{E,i}) = 0 \quad (4.20)$$

Den Ausdruck können wir umschreiben zu

$$\vec{n}_{E,i} \vec{t}_{ES} = c_i \quad (4.21)$$

mit

$$c_i = -\vec{n}_{E,i}(R_{ES} \vec{p}_{S,i} - \vec{p}_{E,i}) \quad (4.22)$$

Da bis auf das gesuchten \vec{t}_{ES} alle Werte bekannt sind, können wir analog zu Gleichung 4.16 ein lineares Gleichungssystem bilden, welches (im Fall von $N > 3$) wieder geschlossen lösbar ist.

Insgesamt erhalten wir eine initiale Schätzung der Transformation zwischen Endeffektor- und Sensorkoordinatensystem.

Modelleinpassung

Das aufgenommene 3D-Bild ist im Allgemeinen verrauscht, es kann Artefakte wie z.B. Reflexionen enthalten, oder auf dem Bild befinden sich außer dem Fußboden weitere Objekte.

Um diese Fehlerquellen zu beseitigen, muss eine Methode zur Identifizierung und Filterung von Ausreißern eingesetzt werden. Es müssen also alle Punkte in der

3D-Aufnahme erkannt werden, die nicht zur Fußbodenebene gehören. Eine gutes Verfahren hierzu ist der bekannte RANSAC-Algorithmus [22]. Der Algorithmus kann in einer Punktemenge Punkte identifizieren, welche bis auf eine vorgegebene Genauigkeit durch ein vorgegebenes Modell beschrieben werden können.

In unserem Fall wird das Modell durch eine Ebenengleichung repräsentiert. Es werden zunächst zufällig drei Punkte ausgewählt. Die Ebene, die durch diese Punkte aufgespannt wird, stellt unsere erste Modellschätzung dar. Danach werden alle Punkte die bis auf einen vorgegebenen Abweichungsfehler in dieser Ebene liegen als Teil dieses Modells betrachtet. Alle übrigen Punkte werden verworfen. Falls die Anzahl der Punkte, die mit dem Modell übereinstimmen, hinreichend groß ist, war der Algorithmus erfolgreich. Andernfalls wird der Vorgang mit drei anderen, zufällig gewählten Punkten so oft wiederholt, bis entweder, die Menge der gültigen Ebenenpunkte groß genug ist oder eine maximale Anzahl an Iterationen überschritten worden ist. Im letzteren Fall konnte im 3D-Bild keine gültige Ebene für die Kalibrierung gefunden werden.

Nichtlineare Optimierung

Das Ergebnis der geschlossenen Lösung wurde durch eine lineare Regression berechnet. Wünschenswert ist jedoch eine direkte Optimierung der freien Parameter der Hand-Auge-Transformation bezüglich einer vorgegebenen Fehlerfunktion. Wir können sechs Parameter identifizieren: Die Translation wird durch die Parameter x , y und z beschrieben, und die Rotation kann durch die drei Euler-Winkel α , β und γ repräsentiert werden.

Die zu minimierende Fehlerfunktion E definieren wir als die Summe der quadratischen Abstände zwischen dem Fußboden und den gemessenen Ebenenpunkten p_S^{ij} . Die Menge $\{p_S^{ij}\}_{j=1..K_i}$ beinhaltet die K_i gültigen Ebenenpunkte der i -ten 3D-Aufnahme. Wir definieren:

$$E(x, y, z, \alpha, \beta, \gamma) = \sum_{i=1}^N \sum_{j=1}^{K_i} \left\| \vec{n}_W(T_{WE}^i T_{ES} p_S^{ij} - p_W^0) \right\|^2 \quad (4.23)$$

Es handelt sich bei der Fehlerfunktion um eine Funktion der Hand-Auge-Transformation parametrisiert durch x, y, z, α, β und γ , wobei N die Anzahl der aufgenommenen 3D-Bilder ist. Die Minimierung dieser Fehlerfunktion stellt ein nicht-lineares Optimierungsproblem dar, welches mit Hilfe von Standardoptimierungsverfahren gelöst werden kann [62].

Nichtlinearen Optimierungsverfahren sind im Allgemeinen lokale Verfahren, d.h. dass vor der Optimierung eine gute Initialisierung der zu schätzenden Para-

	x [mm]	y [mm]	z [mm]	α [°]	β [°]	γ [°]	Fehler [mm ²]
initiale Kalibr.	-153.87	-64.68	194.44	-93.29	4.75	-171.83	552.01
optimierte Kalibr.	-172.57	-15.43	181.04	-92.78	4.31	-170.86	28.73

Tabelle 4.3: Ergebnisse der Hand-Auge-Kalibrierung nach der initialen Schätzung und nach der nichtlinearen Optimierung

meter notwendig ist. Diese Schätzung erhalten wir aus dem zuvor beschriebenen, geschlossenen Ansatz.

Experimentelle Ergebnisse

In diesem Abschnitt werden experimentelle Ergebnisse für das beschriebene Hand-Auge-Kalibrierungsverfahren vorgestellt.

Tabelle 4.3 zeigt die Schätzung der Parameter x , y , z , α , β und γ nach einem Kalibrierungsdurchgang. Es sind sowohl die Ergebnisse der geschlossenen Lösung dargestellt, als auch die Ergebnisse nach der nichtlinearen Optimierung.

Der Fehlerwert (siehe Gleichung 4.23) wird nach dem Optimierungsschritt deutlich kleiner und liegt mit $28.73\text{mm}^2 \approx (5.4\text{mm})^2$ ungefähr im Bereich der Messgenauigkeit des verwendeten PMD-Sensors.

Tabelle 4.4 vergleicht die Ergebnisse von neun verschiedenen Kalibrierungsdurchgängen. Jeder Kalibrierungsdurchgang unterscheidet sich durch verschiedene Sensorpositionen und -orientierungen bei den jeweiligen Aufnahmen. Die geringe Standardabweichung über die verschiedenen Kalibrierungsergebnisse zeigt die Reproduzierbarkeit des vorgestellten Kalibrierungsverfahrens.

4.2 Containerlokalisierung und Clipping

4.2.1 Ansatz zur Bestimmung der Containerposition

In realistischen Szenarien kann nicht davon ausgegangen werden, dass sich der Container immer an der exakt gleichen Position befindet. Die genaue Kenntnis der Containerposition ist jedoch aus zwei Gründen wichtig. Zum einen muss die Position des Containermodells im Szenenmodell mit der realen Position übereinstimmen, so dass eine vom Bahnplaner als kollisionsfrei angenommene Roboterbewegung auch tatsächlich kollisionsfrei ist. Zum anderen wird die Bestimmung

	x [mm]	y [mm]	z [mm]	α [°]	β [°]	γ [°]	Fehler [mm ²]
Kalibr. 1	-171.03	-19.05	181.75	-92.66	3.97	-170.57	26.33
Kalibr. 2	-170.65	-11.62	184.17	-92.53	4.31	-170.88	25.00
Kalibr. 3	-165.72	-18.55	191.01	-92.68	3.93	-170.69	33.41
Kalibr. 4	-172.28	-7.83	191.86	-91.64	4.56	-170.88	24.03
Kalibr. 5	-166.75	-12.40	192.96	-92.07	4.58	-170.40	24.70
Kalibr. 6	-165.66	-17.72	196.81	-92.95	4.89	-170.99	17.52
Kalibr. 7	-170.51	-17.17	193.61	-91.97	4.41	-170.87	21.11
Kalibr. 8	-171.20	-16.65	194.60	-92.40	4.56	-170.94	19.04
Kalibr. 9	-172.93	-11.86	195.71	-92.40	4.97	-171.12	18.40
Std.abw.	1.68	1.98	2.27	0.64	0.60	0.47	2.23

Tabelle 4.4: Die Ergebnisse für neun verschiedene Kalibrierungsdurchgänge liegen nahe beieinander, was sich in einer geringen Standardabweichung ausdrückt.

der Greifpunkte erheblich vereinfacht, da mit bekannter Containerposition einfach zwischen dem Behälter und seinem Inhalt unterschieden werden kann. So kann der Suchraum möglicher Greifpunkte von vornherein auf den Bereich innerhalb des Behälters eingeschränkt werden.

Zur Bestimmung der Containerposition und -orientierung wird ein Modell der Vorderseite des Containers benötigt (siehe Abb. 4.8). Dieses Modell wird mit Hilfe eines iterativen Einpassungsverfahrens in die zuvor aufgenommene 3D-Punktwolke eingepasst (siehe Abb. 4.9). Als Ergebnis erhalten wir die gesuchte Transformation zwischen Welt- und Containerkoordinatensystem (das genaue Verfahren wird in Abschnitt 4.3 erläutert).

Mit dem Containermodell ist zusätzlich ein sogenanntes Clipping-Volumen durch eine feste Transformation verknüpft. Dieses Clipping-Volumen ist ein virtueller Körper, der die gleiche Form wie der Container besitzt, jedoch etwas kleinere Abmessungen hat und den Bereich des Containerinneren umfasst. Werden alle Punkte verworfen (clipping), die sich außerhalb dieses Körpers befinden, erhält man eine reduzierte Punktwolke, welche nur noch den zu entladenden Inhalt repräsentiert. Der Vorgang des Clippings ist in Abbildung 4.10 dargestellt.

4.2.2 Experimente zur Containerlokalisierung

In Tabelle 4.5 sind die Ergebnisse einiger Versuche zur Bestimmung der Containerposition dargestellt. Mit Hilfe eines Messtools wurde zunächst die Position des Containers und seine Verdrehung um die Hochachse bestimmt. Anschließend

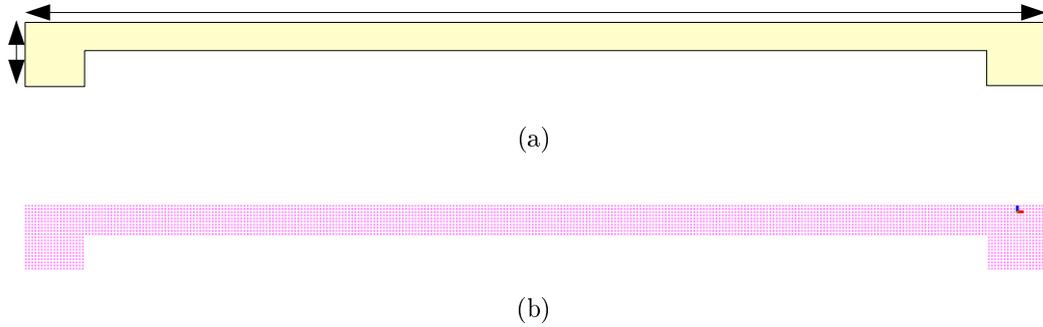


Abbildung 4.8: Containermodell. (a) Für die Containerlokalisierung wird ein Teil des Containerrahmens modelliert. (b) Der Einpassungsalgorithmus benötigt das Modell als Punktwolke.

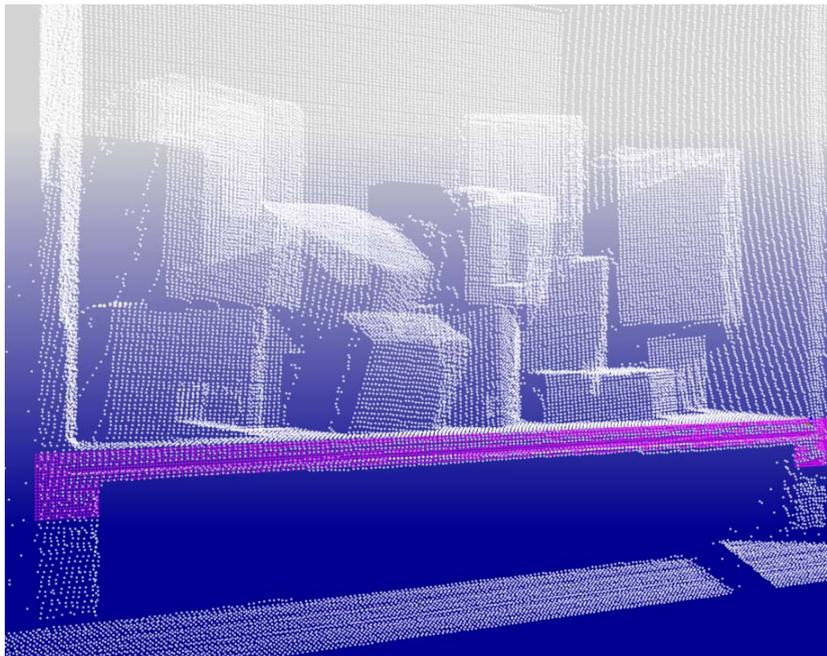


Abbildung 4.9: Nach erfolgreicher Einpassung befindet sich das Containermodell an der richtigen Position in der 3D-Punktwolke.

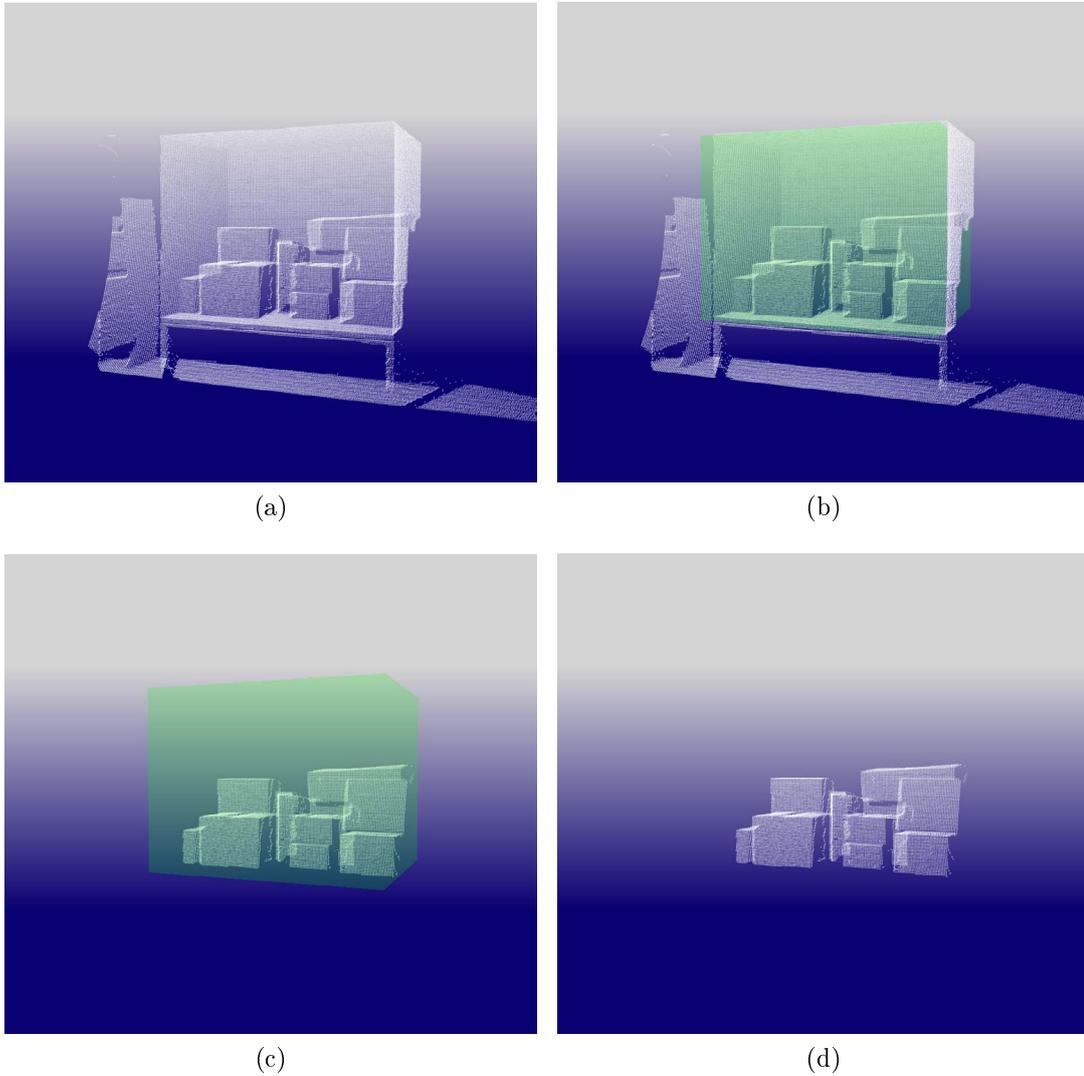


Abbildung 4.10: Clipping der 3D-Punktwolke. (a) Vollständige 3D-Punktwolke einer Szene, (b) Erkannte Position des Behälters, (c) Ausschneiden des Behälterinhalts, (d) Reduzierte Punktwolke.

initiale Position x, y, z, α [mm, °]				Ergebnis x, y, z, α [mm, °]				Fehler [mm ²]	Iterationen
0	0	0	0	-0.092	-0.582	0.449	0.086	23.44	10
-50	0	0	0	-0.100	-1.183	0.371	0.088	23.38	23
-100	0	0	0	-0.100	-1.183	0.371	0.088	23.38	30
0	-50	0	0	0.300	-7.917	2.417	0.107	23.53	84
0	50	0	0	0.079	1.310	2.369	0.092	23.32	100
0	0	0	5	-0.059	-3.295	0.325	0.090	23.34	20
0	0	0	-5	0.300	-7.917	2.417	0.107	23.53	92
70	0	0	5	0.124	-2.314	2.486	0.095	22.98	30
-70	0	0	-5	0.161	-3.237	2.484	0.098	22.95	22

Tabelle 4.5: Ergebnisse der Containerlokalisierung

wurde eine 3D-Punktwolke des Containers aufgenommen. Für die Versuche wurde der Einpassungs-Algorithmus mit verschiedenen Startwerten initialisiert und solange iteriert bis dieser zu einem Ergebnis konvergiert ist. Die Tabelle zeigt die ermittelten Positionen und Orientierungen bei verschiedenen Startwerten. Die angegebenen Positions- und Winkelwerte sind relativ zum eingemessenen Containerkoordinatensystem, d.h. bei einer Position von $(0, 0, 0)$ und einem Winkel von 0° wären eingemessene und geschätzte Position in Übereinstimmung.

Wie zu erkennen ist, konvergiert der Algorithmus auch bei einer Initialisierung von $(0, 0, 0, 0)$ zu einem lokalen Minimum, welches nicht mit den gemessenen Werten übereinstimmt. Diese Abweichung beruht wahrscheinlich auf Ungenauigkeiten bei der Vermessung des Containers.

4.3 Beschreibung des ICP-Verfahrens

4.3.1 Das Verfahren

Das „Iterative Closest Point“-Verfahren (ICP-Verfahren) ist ein allgemeines Verfahren, um zwei geometrische Modelle so auszurichten, dass ein bestimmtes Abstandsmaß minimiert wird [6, 14]. Es existieren verschiedene Varianten des ICP-Verfahrens [65], die sich u.a. durch die Repräsentation der geometrischen Modelle und der Art der Fehlerminimierung unterscheiden.

Im Folgenden soll ein ICP-Algorithmus vorgestellt werden, bei dem zwei 3D-Punktwolken so ausgerichtet werden, dass der mittlere quadratische Fehler zwischen korrespondierenden Punkten minimiert wird. Das Ergebnis ist eine Trans-

formation, welche die eine Punktwolke in die andere überführt.

Gegeben seien zwei Punktemengen $P = \{p_1 \cdots p_{N_P}\}$ und $M = \{m_1 \cdots m_{N_M}\}$, welche z.B. eine aufgenommene Punktwolke und ein einzupassendes Modell repräsentieren können. Für unsere Anwendungen gehen wir von 3D-Punkten (später von 2D-Punkten) aus. Das Verfahren selbst gibt jedoch keine Beschränkung hinsichtlich der Dimensionalität vor.

Gesucht ist eine Translationvektor \vec{t} und eine Rotationsmatrix R , welche die Fehlerfunktion

$$E(R, \vec{t}) = \sum_{i=1}^{N_M} \|q_i - R\vec{m}_i - \vec{t}\|^2 \quad (4.24)$$

minimiert. Dabei ist $Q = \{q_i\}$ die Menge der Punkte, die man erhält, wenn für jeden transformierten Punkt aus M der nächste Nachbarpunkt in P bestimmt wird:

$$q_i = \arg \min_{p_j} \|p_j - R\vec{m}_i - \vec{t}\| \quad , \text{ für } i = 1 \cdots N_M \quad (4.25)$$

Da die Gleichungen 4.24 und 4.25 jeweils voneinander abhängig sind, existiert keine geschlossene Lösung, welche die beiden Gleichungen gleichzeitig löst. Der ICP-Algorithmus geht deshalb iterativ vor. Zunächst wird eine initiale Transformation $(R^{(0)}, \vec{t}^{(0)})$ vorgegeben. Entsprechend Gleichung 4.25 werden die nächsten Nachbarn $\{q_i\}$ zu den Punkten aus M unter der aktuellen Transformation bestimmt und man erhält zwei korrespondierende Punktemengen. Sind diese beiden Punktemengen erst einmal gegeben, lässt sich 4.24 geschlossen lösen. In [21] werden verschiedene Lösungsmöglichkeiten für dieses Problem vorgestellt und miteinander verglichen. In [2] wird eine Lösungsmöglichkeit auf Basis einer Korrelationsmatrix vorgestellt. Zunächst werden die Mittelwerte der beiden Punktemengen von den Punkten subtrahiert:

$$\bar{q}_i = q_i - \bar{q} \quad \text{mit} \quad \bar{q} = \frac{1}{N_M} \sum_{i=1}^{N_M} q_i \quad (4.26)$$

und

$$\bar{m}_i = m_i - \bar{m} \quad \text{mit} \quad \bar{m} = \frac{1}{N_M} \sum_{i=1}^{N_M} m_i \quad (4.27)$$

Anschließend wird die Korrelationsmatrix H berechnet:

$$H = \sum_{i=1}^{N_m} \bar{m}_i \bar{m}_i^T \quad (4.28)$$

```

1: procedure ICP(model, pointcloud,  $(R^{(0)}, t^{(0)})$ )
2:    $E \leftarrow \infty$ 
3:    $R \leftarrow R^{(0)}$ 
4:    $t \leftarrow t^{(0)}$ 
5:   repeat
6:      $E_{\text{old}} \leftarrow E$ 
7:     model  $\leftarrow$  transform( model,  $(R, t)$  )
8:     pointcloud'  $\leftarrow$  findNearestNeighbours( pointcloud , model)
9:      $(R, t) \leftarrow$  estimateNewTransformation(pointcloud', model)
10:     $E \leftarrow$  costFunction(pointcloud', model)
11:  until  $E_{\text{old}} - E < \epsilon$ 
12:  return  $(R, t)$ 
13: end procedure

```

Abbildung 4.11: Der ICP-Algorithmus nach Besl und McKay [6] als Pseudocode.

Ist $UAV^T = H$ die Singulärwertzerlegung von H , so gilt für die optimale Rotation

$$\hat{R} = VU^T \quad (4.29)$$

und für die optimale Translation

$$\hat{t} = \bar{q} - \hat{R}\bar{m} \quad (4.30)$$

In Gleichung 4.29 kann die Determinante von \hat{R} in bestimmten Fällen -1 ergeben. Dann gilt stattdessen:

$$\hat{R} = V \begin{pmatrix} 1 & & \\ & 1 & \\ & & \det(UV^T) \end{pmatrix} U^T \quad (4.31)$$

Die Schritte Nächste-Nachbar-Suche, Aktualisierung der Rotation und der Translation, und Transformation der Punktwolke werden so lange wiederholt, bis die Änderung der Fehlerfunktion E unter einen vorgegebenen Schwellwert fällt. In Abbildung 4.11 ist der ICP-Algorithmus noch einmal als Pseudocode zusammengefasst.

4.3.2 Der kd-Tree

Für eine effiziente Implementierung des ICP-Algorithmus ist es entscheidend, wie schnell in einer Punktwolke für einen vorgegebenen Punkt der nächste Nach-

barpunkt gefunden werden kann. Für diese Aufgabe bietet sich der sogenannte kd-Tree an [4].

Ein kd-Tree ist eine Datenstruktur zur Speicherung von Punkten aus einem k -dimensionalen Raum. Bei einem kd-Tree handelt es sich um einen binären Suchbaum, in dem jeder Punkt durch einen Knoten repräsentiert wird. Jeder Knoten teilt die Menge der nachfolgenden Punkte in zwei Teilmengen, die entlang einer (Hyper-)Ebene geteilt sind, wobei die Ebenen stets senkrecht zu einer der Raumachsen orientiert sind. Eine wichtige Eigenschaft des kd-Trees besteht darin, dass in ihm effizient nach dem nächsten Nachbarn zu einem gegebenen Punkt gesucht werden kann. Der amortisierte Aufwand für die Nächste-Nachbar-Suche beträgt $O(\log(N))$, wobei N die Anzahl der Punkte ist.

4.4 Greifpunktbestimmung in Punktwolken

4.4.1 Greifflächenmodell

Die Idee für die Bestimmung geeigneter Greifpunkte in einer 3D-Punktwolke besteht darin, nach Flächen zu suchen, an denen ein vorgegebener Greifertyp ansetzen kann. Dies geschieht durch das Einpassen eines zuvor erstellten Modells in die Punktwolke. Das Modell richtet sich dabei nach dem verwendeten Greifer und bildet im Wesentlichen die Kontaktfläche zwischen Greifer und Objekt nach. Abbildung 4.12 zeigt die Modelle für einen kreisförmigen Sauggreifer und einen ovalen Flachgreifer.

Vakuumgreifer erzeugen die notwendige Kraft zum Heben eines Objektes durch Unterdruck. Deshalb muss der Greifer möglichst bündig mit der Objektoberfläche abschließen. Für den ersten Greifer ist dazu eine ebene Fläche notwendig, während der zweite Greifer auch an hervorstehenden Bereichen ansetzen kann, sofern der Rand des Greifers genügend Kontakt hat.

Die Geometrie der beiden Modelle ist durch wenige Parameter veränderbar, so dass die Modelle leicht an verschiedene Greiferabmessungen angepasst werden können.

4.4.2 Finden der Greifpunkte

Zur Bestimmung möglicher Greifpunkte wird das Greiferflächenmodell, welches in Form einer Punktwolke vorliegt, an zufälligen Positionen in der zuvor aufgenommenen und beschnittenen 3D-Punktwolke platziert. Ausgehend von den

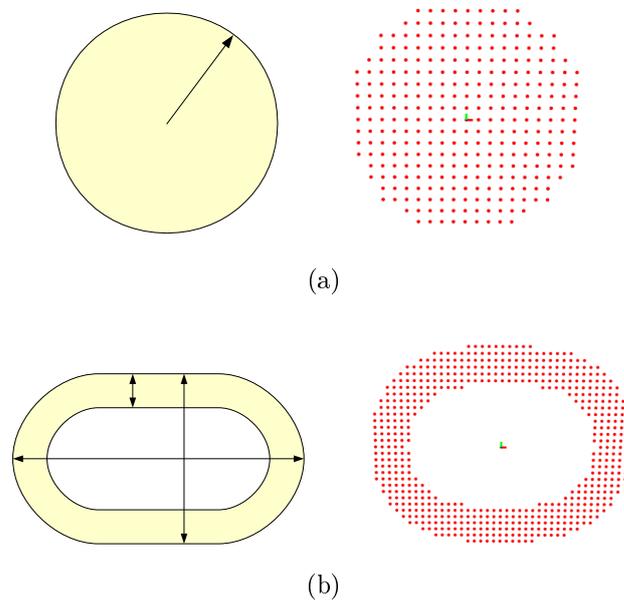


Abbildung 4.12: Modellierung der Greiferkontaktflächen für zwei Vakuumgreifer. Aus den geometrischen Modellen der Greifer werden Punktwolken für den ICP-Algorithmus abgeleitet. (a) kreisförmiger Vakuumgreifer, (b) ovaler Vakuumgreifer.

initialen Positionen wird das Modell mit Hilfe des ICP-Verfahrens in die Punktwolke eingepasst (siehe Abb. 4.13).

Das Greifflächenmodell ist mit einem festen, lokalen Koordinatensystem verknüpft. Nach erfolgreicher Einpassung kann aus der Transformation dieses Koordinatensystems die Position und die Orientierung eines möglichen Greifpunktes abgelesen werden.

Abbildung 4.14 zeigt mehrere in einer 3D-Punktwolke gefundene Greifpunkte, repräsentiert durch ihre lokalen Koordinatensysteme.

In Abbildung 4.15 ist der Algorithmus für die Bestimmung der Greifpunkte als Pseudocode dargestellt. Vorgegeben werden neben der Punktwolke (point-cloud) und dem Greifflächenmodell (model) eine maximale Anzahl an Iterationen (maxGuesses), ein maximal zulässiger Einpassungsfehler (maxFittingError) und eine initiale Orientierung (initialOrientation). In jeder Iteration wird die Position t mit den Koordinaten eines zufällig gewählten Punktes der Punktwolke und die Orientierung R mit der initialen Orientierung initialisiert. Anschließend wird das Greifflächenmodell mit Hilfe des ICP-Verfahrens in die Punktwolke eingepasst, d.h. der mittlere quadratische Abstand zwischen Modell und Punk-

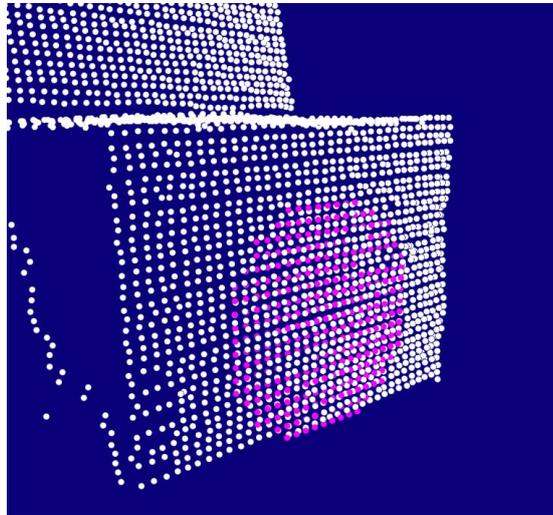


Abbildung 4.13: Mit Hilfe des ICP-Verfahrens wird das Greiferflächenmodell in die 3D-Punktwolke eingepasst.

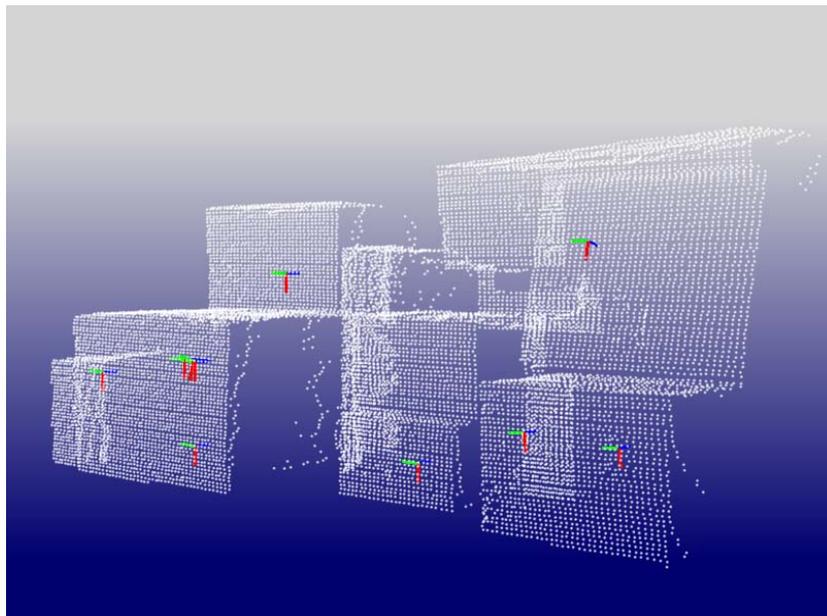


Abbildung 4.14: In der 3D-Punktwolke sind zehn Greifpunkte bestimmt worden. Ihre Positionen und Orientierungen sind durch lokale Koordinatensysteme repräsentiert.

```

1: procedure GREIFPUNKTBESTIMMUNG(model, pointcloud)
2:   for  $i \leftarrow 1$  to maxGuesses do
3:      $t \leftarrow$  chooseRandomPoint(pointCloud)
4:      $R \leftarrow$  initialOrientation
5:     fittingError  $\leftarrow$  ICP(model, pointcloud, ( $R, t$ ))
6:     if fittingError < maxFittingError then
7:       addToGraspPositionList(( $R, t$ ))
8:     end if
9:   end for
10:  return GraspPositionsList
11: end procedure

```

Abbildung 4.15: Greifpunktbestimmung mit ICP-Verfahren als Pseudocode.

tewolke (fittingError) minimiert. Falls dieser Fehlerwert kleiner als der maximal zulässige Einpassungsfehler ist, wird der gefundene Greifpunkt der Liste potentieller Greifpunkte hinzugefügt (addToGraspPositionList), ansonsten wird er als ungültig verworfen. Da es sich beim ICP-Verfahren um ein lokales Optimierungsverfahren handelt, konvergiert der Algorithmus immer in ein lokales Minimum, welches sich in der Nähe der initialen Greifposition befindet.

4.4.3 Ergebnisse

Abbildung 4.16 zeigt die Häufigkeitsverteilung der Einpassungsfehler zwischen Greifermodell und 3D-Punktwolke für gefundene Greifpunkte. Jeder Balken des Histogramms repräsentiert die Anzahl der Greifpunkte, deren Fehlerwerte in einem Intervall von 10 mm^2 liegen, mit Ausnahme des letzten Balkens, der die Anzahl aller Greifpunkte mit einem Fehlerwert über 400 repräsentiert.

Für den Versuch wurden insgesamt 1000 Greifpunkte nach dem vorgestellten Verfahren ermittelt. In der Abbildung sind alle Greifpunkte dargestellt, auch solche mit sehr hohen Fehlerwerten. In der untersuchten Anwendung wurde ein empirischer Schwellwert (maxFittingError) von 50 mm^2 ermittelt, bis zu dem Greifpunkte als gültig betrachtet werden, d.h. in die Liste potentieller Greifpunkte übernommen werden. Greifpunkte mit einem größeren Fehlerwert werden verworfen. Daraus ergibt sich eine Rate von 57,8% gültiger Greifpunkte bezügliche aller bestimmten Greifpunkte.

In Abbildung 4.17 ist die Verteilung der benötigten Zeiten für die Bestimmung

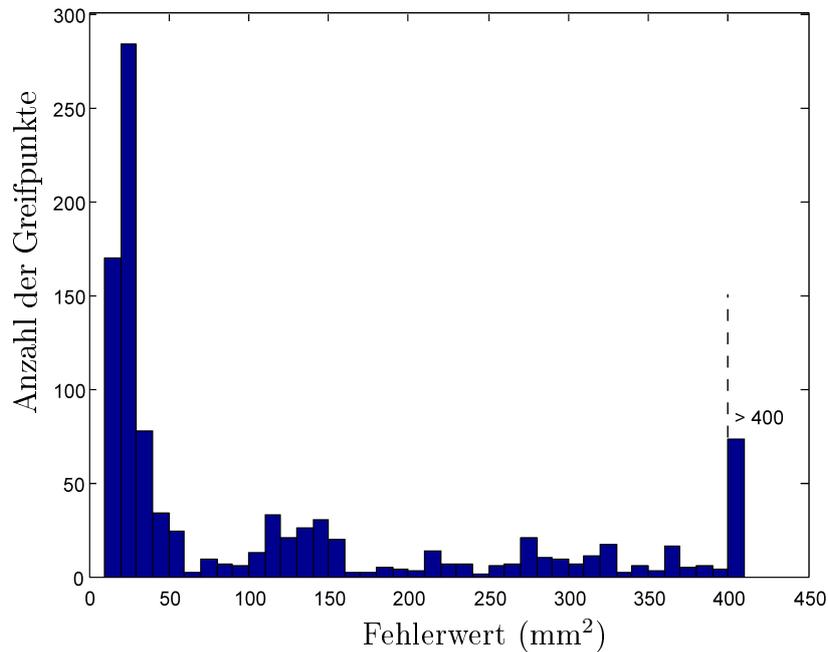


Abbildung 4.16: Häufigkeitsverteilung der Einpassungsfehler für gefundene Greifpunkte (Alle Werte über 400 wurden zusammengefasst).

eines Greifpunktes dargestellt. Die angegebenen Laufzeiten wurden auf einem Intel Core 2 mit 2GHz ermittelt, die Punktwolke hatte einen Umfang von 12.814 Punkten und der ICP-Algorithmus war auf maximal 30 Iterationen begrenzt. Es ergab sich eine mittlere Laufzeit von 280 Millisekunden für die Bestimmung eines Greifpunktes.

4.5 Greifpunktbestimmung in Tiefenbildern

4.5.1 Ansatz

Zur Bestimmung von Greifpunkten für Zweifingergreifer (Parallelbacken- und Winkelgreifer) wurde ein Verfahren entwickelt, das auf Tiefenbildern arbeitet. Ähnlich wie bei der Greifpunktbestimmung in 3D-Punktwolken, besteht die Grundidee in der Einpassung eines greiferabhängigen Modells in das zuvor aufgenommene Tiefenbild.

Im Folgenden wird zuerst die Gewinnung und Optimierung des Tiefenbildes beschrieben, anschließend wird die Bestimmung der Greifpunkte mit Hilfe eines 2D-ICP-Ansatzes erläutert.

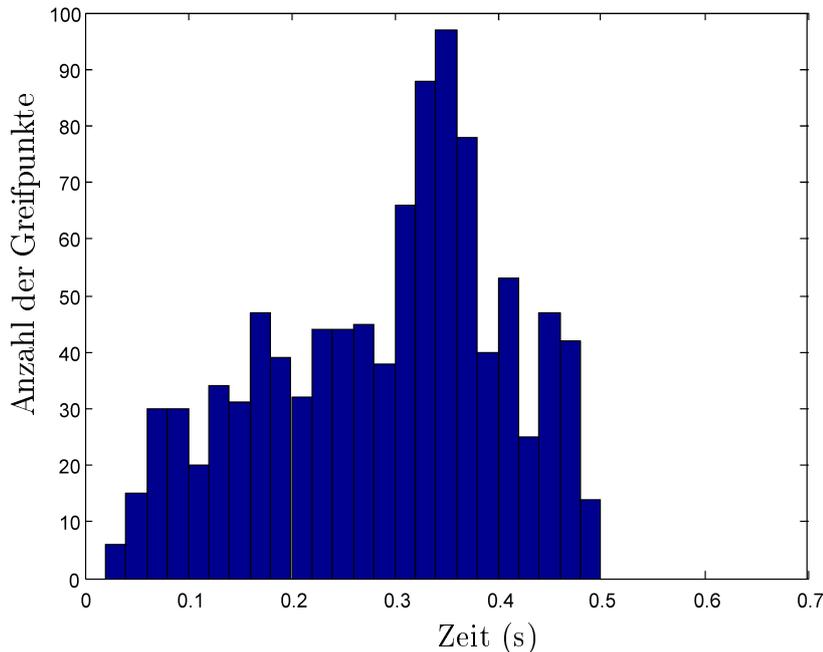


Abbildung 4.17: Häufigkeitsverteilung der Zeiten für die Bestimmung eines Greifpunktes

4.5.2 Das Tiefenbild

Ein Tiefenbild ist eine Abbildung $f : \Omega \rightarrow \mathbb{R}$ bei der jedem 2D-Punkt $(i, j) \in \Omega$ ein Höhenwert k zugeordnet wird. In Zusammenhang mit Tiefenbildern wird häufig auch von 2,5D-Daten gesprochen.

Zum Erstellen der Tiefenbilder gehen wir zunächst von einer zuvor aufgenommenen 3D-Punktewolke aus (siehe Abb. 4.19).

Durch die Punktewolke wird eine Schnittebene gelegt, über die ein Tiefenbild mit $\Omega = [0, \dots, w - 1] \times [0, \dots, h - 1]$ aufgespannt wird, wobei w die Breite und h die Länge des Tiefenbildes in Pixeleinheiten ist. Die Position der Schnittebene in Weltkoordinaten wird durch eine Transformation $T_W^F = (R|t)$ festgelegt. Da das Tiefenbild nur an diskreten 2D-Koordinaten festgelegt ist, benötigen wir zusätzlich einen Diskretisierungsfaktor s . Abbildung 4.18 zeigt den Algorithmus zur Umwandlung einer 3D-Punktewolke in ein Tiefenbild als Pseudocode.

Da die Punkte in der Punktewolke dünn besetzt sind, existieren im Tiefenbild Positionen, für die keine Tiefenwerte definiert sind (NaN, Not a Number). Diese Stellen müssen durch Interpolation aufgefüllt werden (siehe Abb. 4.19(c)). Um dies zu erreichen, wird das Verfahren der gewichteten Mittelung [24] angewendet. Bei dieser, auch als normalisierte Faltung bezeichneten Operation wird das

```

1: procedure TIEFENBILD(pointcloud,  $T_F^W$ ,  $w$ ,  $h$ ,  $s$ )
2:    $f(i, j) \leftarrow \text{NaN}, \forall (i, j) \in \Omega$ 
3:   for each  $(x, y, z) \in \text{pointcloud}$  do
4:      $(x', y', z', 1)^T \leftarrow (T_F^W)^{(-1)}(x/s, y/s, z, 1)^T$ 
5:      $(i, j, k) \leftarrow (\lfloor x' \rfloor, \lfloor y' \rfloor, \lfloor z' \rfloor)$ 
6:     if  $0 \leq i < w$  and  $0 \leq j < h$  and  $k > f(i, j)$  then
7:        $f(i, j) \leftarrow k$ 
8:     end if
9:   end for
10:  return
11: end procedure

```

Abbildung 4.18: Die Tiefenbildgenerierung als Pseudocode

Tiefenbild mit einer Gewichtungsmatrix multipliziert und anschließend mit einer Filtermaske gefaltet und wieder normiert. Die Gewichtungsmatrix w enthält in unserem Fall nur die Einträge 1 und 0, für gültige bzw. undefinierte Tiefenwerte:

$$w(i, j) = \begin{cases} 0 & \text{falls } f(i, j) = \text{NaN} \\ 1 & \text{sonst} \end{cases} \quad (4.32)$$

Daneben benötigen wir eine Faltungsmaske h , welche als Glättungsfilter verwendet wird. Wir benutzen einen zweidimensionalen Binomialfilter, der für eine Maskengröße von 3×3 folgendes Aussehen hat:

$$h = \frac{1}{4} [1 \quad 2 \quad 1] * \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} = \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} \quad (4.33)$$

Binomialfilter haben die nützliche Eigenschaft, dass ihre Filtermasken separierbar sind, d.h. sie lassen sich entlang der horizontalen und vertikalen Achse in eindimensionale Filtermasken zerlegen, was für eine effiziente Implementierung vorteilhaft ist. Für das interpolierte Tiefenbild f' ergibt sich

$$f' = \frac{h * (w \cdot f)}{h * w} \quad (4.34)$$

Dabei bezeichnet $*$ die (diskrete) Faltung. Die Multiplikation und die Division sind elementweise auszuführen. Auf das so gewonnene vollständige Tiefenbild f'

wird anschließend ein Medianfilter angewendet. Der Medianfilter zeichnet sich dadurch aus, dass er effektiv Rauschen unterdrücken kann, während Kanten erhalten bleiben. Das Ergebnis dieser beiden Filterschritte ist in Abbildung 4.20(a) dargestellt.

4.5.3 Kantendetektion und 2D-ICP

Im Gegensatz zu einem Vakuumgreifer, der für einen sicheren Griff eine ebene Kontaktfläche benötigt, sind es bei einem Backengreifer zwei gegenüberliegende Kontaktflächen. Um geeignete Kontaktflächen zu finden, wird das Tiefenbild nach signifikanten Höhenänderungen durchsucht, indem eine Kantendetektion durchgeführt wird. Dazu wird das Tiefenbild partiell in x - und y -Richtung abgeleitet. Die diskreten Approximationen g_x und g_y der Ableitungen erhalten wir durch Anwendung des Sobeloperators. Dies lässt sich als Faltung eines Tiefenbildes f mit den entsprechenden Filtermasken M_x und M_y ausdrücken:

$$g_x = f * M_x \quad (4.35)$$

$$g_y = f * M_y \quad (4.36)$$

mit

$$M_x = \begin{pmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{pmatrix} \quad \text{und} \quad M_y = \begin{pmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{pmatrix}$$

Aus g_x und g_y erhalten wird dann das Gradientenstärkebild g_{mag} und das Gradientenrichtungsbild g_{dir} :

$$g_{mag} = \sqrt{g_x g_x + g_y g_y} \quad (4.37)$$

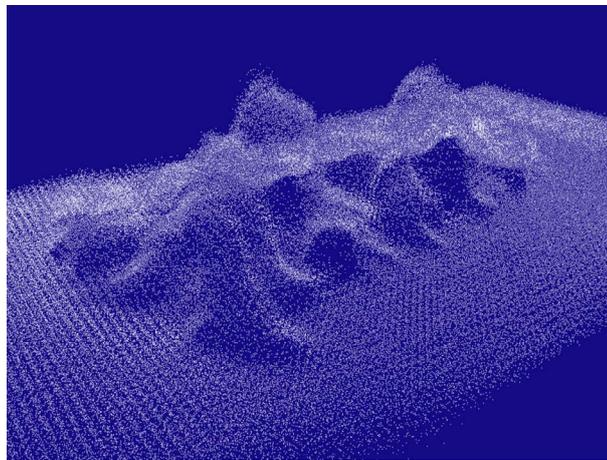
$$g_{dir} = \arctan2(g_y, g_x) \quad (4.38)$$

Eine Binarisierung des Gradientenstärkebildes anhand eines Schwellwertes würde zwar schon ein Kantenbild ergeben, allerdings wären die Kanten im Allgemeinen mehrere Pixel breit, so dass ihre genaue Lage nicht unmittelbar bekannt ist. Werden nur Pixel als Kantenpunkte ausgewählt, welche entlang der Gradientenrichtung ein Maximum bilden, so erhalten wir den exakten, einen Pixel breiten Verlauf der Kante (Abb. 4.20 (b)). Dieses Verfahren wird auch als „non-maximum suppression“ bezeichnet. Die vorgestellte Methode zur Detektion von Kanten ist im Wesentlichen eine vereinfachte Version des sogenannten Canny-Algorithmus [13].

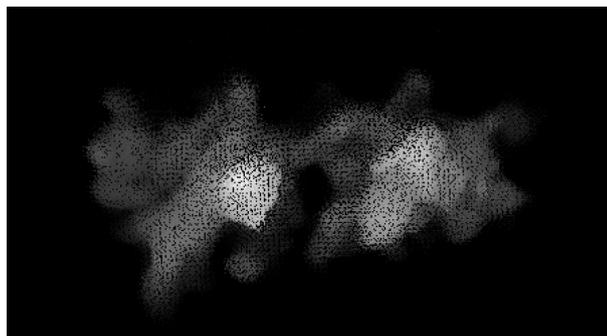
Für die Greifpunktebestimmung wird aus dem Kantenbild eine 2D-Punktmenge erzeugt, indem für jeden Kantenpixel ein Punkt mit den entsprechenden



(a)

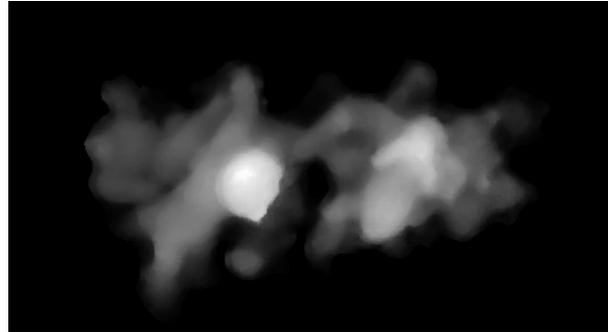


(b)

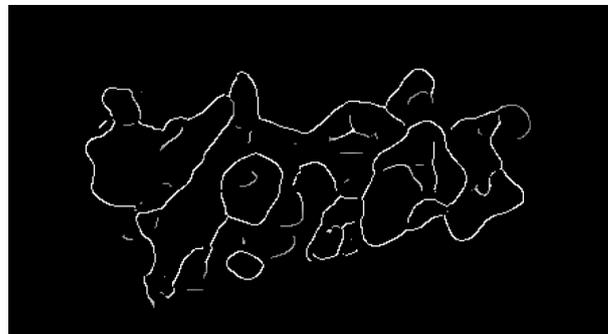


(c)

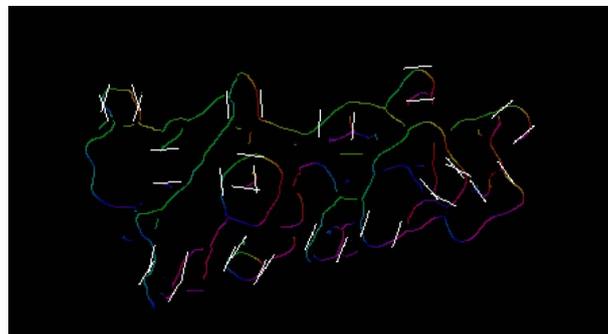
Abbildung 4.19: Berechnung des Tiefenbildes. (a) Szene mit verschiedenen Stofftieren, (b) aufgenommene 3D-Punktewolke, (c) Tiefenbild mit nicht definierten Werten.



(a)



(b)



(c)

Abbildung 4.20: Berechnung von Greifpunkten in einem Tiefenbild. (a) Vervollständigtes und geglättetes Tiefenbild, (b) Gradientenstärkebild (nach non-maximum suppression), (c) Gradientenrichtungsbild mit gefundenen Greifpunkten (die Gradientenrichtung ist farblich codiert, die weißen Linien zeigen die Greiferbacken an).

x/y -Koordinaten generiert und der 2D-Punktemenge hinzugefügt wird. Zusätzlich wird für jeden Punkt der Winkel der Gradientenrichtung an der jeweiligen Position abgespeichert (Abb. 4.21 (a)).

Das greiferabhängige Modell liegt ebenfalls als eine Menge von 2D-Punkten mit vorgegebenen Gradientenrichtungen vor. Im Falle eines Backengreifers besteht das Greifermodell aus zwei parallelen Punktereihen, deren Abstand voneinander mit der maximalen Öffnungsbreite des verwendeten Greifers übereinstimmt (Abb. 4.21 (b)).

Das Einpassen des Modells in die 2D-Punktemenge geschieht wieder mit dem bekannten ICP-Verfahren (siehe Abschnitt 4.3). Bei der Anwendung im Tiefenbild verwenden wir jedoch eine modifizierte 2D-Version des Algorithmus: Hierbei werden nur noch drei Freiheitsgrade geschätzt, nämlich die Translation in x - und y -Richtung, sowie der Drehwinkel um die z -Achse. Des Weiteren ersetzen wir die Nächste-Nachbar-Suche durch die in Abbildung 4.22 dargestellte Prozedur. Für die Abstandsberechnung zwischen einem Kanten- und einem Modellpunkt wird neben der euklidischen Distanz auch der Differenzwinkel zwischen den Gradientenrichtungen berücksichtigt. Eine notwendige Bedingung für nächste Nachbarschaft ist ein Differenzwinkel von maximal 45° .

Abbildung 4.21 zeigt den Vorteil der neuen Abstandsfunktion. Würde man die Gradientenrichtung nicht berücksichtigen, erhielte man einen geringen Einpassungsfehler, wenn sich das Greifermodell zwischen den beiden Objekten befinden würde.

Das weitere Vorgehen bei der Greifpunktbestimmung in Tiefenbildern ist ähnlich zu dem in Abschnitt 4.4 beschriebenen Verfahren: Das Modell wird mehrmals mit zufälligen Positionen und Orientierungen initialisiert und mit Hilfe des 2D-ICP-Verfahrens eingepasst. Ist der Einpassungsfehler nach der Optimierung hinreichend klein, werden die berechnete Translation und Drehung der Liste potentieller Greifpunkte hinzugefügt. Abbildung 4.20(c) zeigt das Ergebnis einer Greifpunktbestimmung in einem Tiefenbild.

4.5.4 Anwendung

Die in Abbildung 4.23 gezeigte Anwendung demonstriert den Einsatz der vorgestellten Greifpunktbestimmung auf Tiefenbildern. Die zu greifenden Stofftiere sind unterschiedlich geformt und flexibel verformbar. Ein Ansatz zur Greifpunktbestimmung auf Basis einer vollständigen Modellierung der Objekte ist daher schwierig. Unser Ansatz dagegen kann ohne Trainings- oder Modellierungsphase die dargestellten Objekte robust greifen.

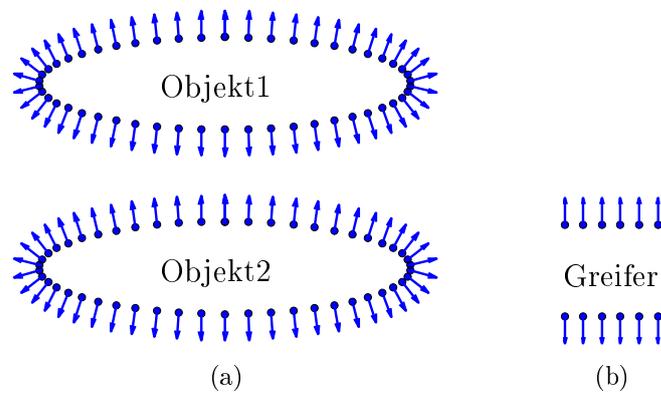


Abbildung 4.21: Greifpunktbestimmung in Tiefenbildern. (a) Aus dem Tiefenbild werden Kantenpunkte, mit dazugehörigen Gradientenrichtungen extrahiert. (b) Das Modell des Parallelbackengreifers besteht aus parallelen Punktereihen.

```

1: procedure FINDNEARESTNEIGHBOURS2D(pointcloud, model)
2:    $E \leftarrow 0$ 
3:   for each  $(x, y, \alpha) \in \text{model}$  do
4:      $d_{min} \leftarrow \infty$ 
5:     for each  $(x', y', \alpha') \in \text{pointcloud}$  do
6:        $d \leftarrow (x - x')^2 + (y - y')^2$ 
7:        $\Delta\alpha \leftarrow \begin{cases} \|\alpha - \alpha'\| & \text{falls } \|\alpha - \alpha'\| < \pi \\ 2\pi - \|\alpha - \alpha'\| & \text{sonst} \end{cases}$ 
8:       if  $d < d_{min}$  and  $\Delta\alpha < \pi/4$  then
9:          $d_{min} \leftarrow d$ 
10:         $(x_n, y_n, \alpha_n) \leftarrow (x', y', \alpha')$ 
11:       end if
12:     end for
13:     if  $d_{min} \neq \infty$  then
14:       addToNearestNeighboursList(  $(x_n, y_n, \alpha_n)$  )
15:     else
16:       return NULL
17:     end if
18:   end for
19:   return nearestNeighboursList
20: end procedure

```

Abbildung 4.22: Die Suche der nächsten Nachbarn in der 2D-Punktemenge als Pseudocode.

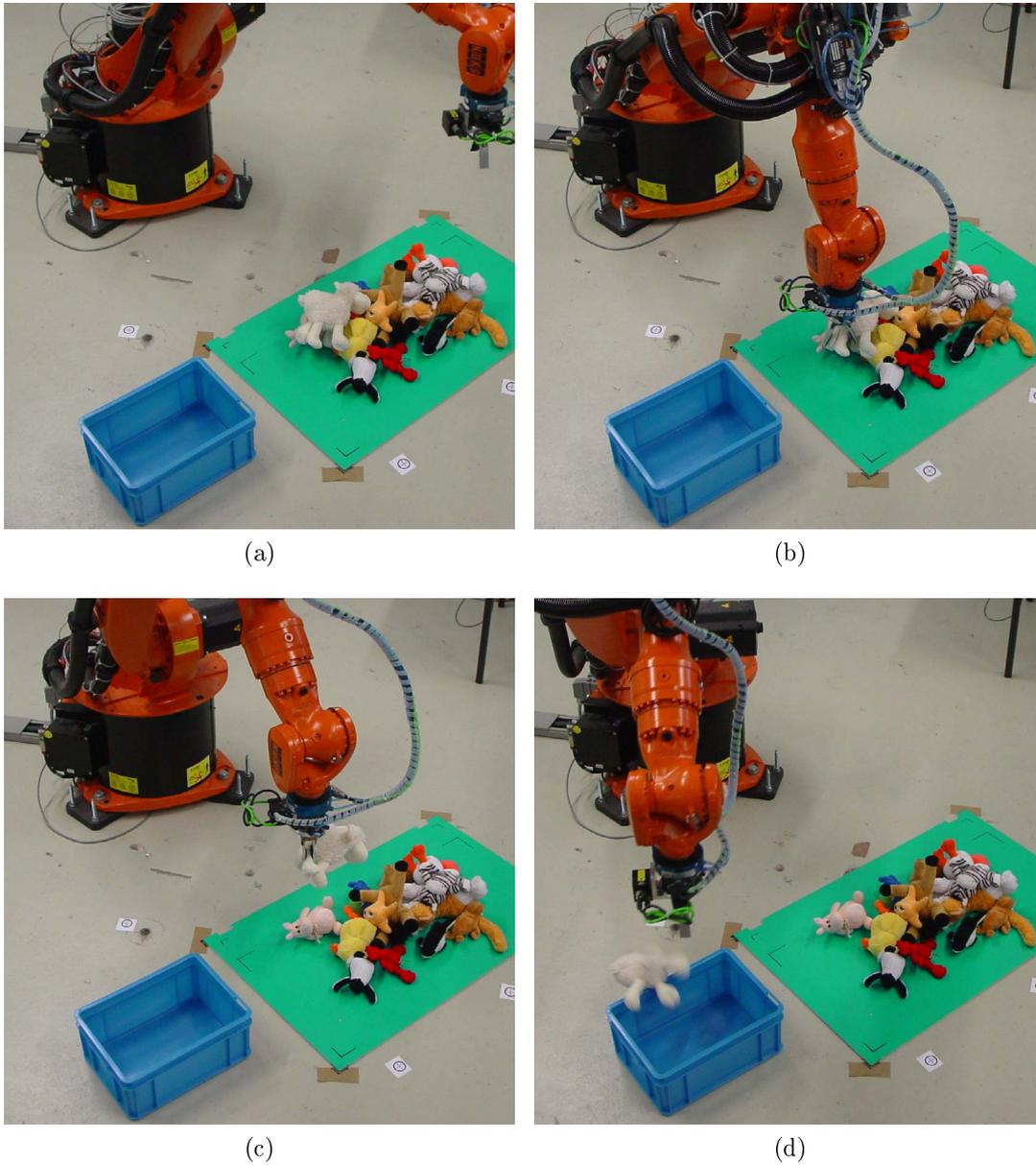


Abbildung 4.23: Greifen von flexiblen Objekten. (a) Die Szene wird durch den PMD-Sensor erfasst. (b) Der Roboter verfährt an die berechnete Greifposition. (c) Das Objekt wird mit Hilfe eines Winkelgreifers gegriffen. (d) Das Objekt wird in einem Behälter abgeladen.

5 Semi-automatisches Entladesystem

Dieses Kapitel beschreibt den semi-automatischen Modus des Entladesystems, bei dem der Nutzer interaktiv in den Entladevorgang eingreifen kann. Es wurden zwei verschiedene Eingabemodalitäten untersucht und in zwei unterschiedlichen Entladeszenarien getestet.

Im Folgenden soll zunächst die Integration eines semi-automatischen Modus, als Ergänzung zum vollautomatischen System motiviert und der allgemeine Ablauf dargestellt werden. Anschließend wird in Abschnitt 5.2 ein System vorgestellt, welches auf einer bildschirmbasierten Mensch-Maschine-Schnittstelle beruht [33, 35]. In Abschnitt 5.3 wird ein neuartiges Laserpointertool beschrieben, durch das der Benutzer mit dem System interagieren kann.

5.1 Einführung

Das Hauptmerkmal des semi-automatischen Entladesystems besteht darin, dass die Auswahl des nächsten zu greifenden Objektes nicht mehr automatisch erfolgt, sondern durch einen Benutzer vorgenommen wird. Das Entladesystem wird durch den Benutzer unterstützt, indem der Mensch seine überlegenen sensorischen und kognitiven Fähigkeiten in das System einbringt.

Ein semi-automatisches bzw. interaktives Entladesystem wird durch drei Ziele motiviert:

1. Produktivsystem: Die Fähigkeiten des Menschen hinsichtlich Objekterkennungsleistungen und Szenenanalyse sind denen von maschinellen Systemen nach aktuellem Stand der Technik bei weitem überlegen. Die Stärken von Robotern liegen in der hohen Geschwindigkeit, in der sie Objekte manipulieren können, in der hohen Positioniergenauigkeit und der Ausdauerleistung. In dem hier vorgestellten System können sich die Stärken von Mensch und Maschine gut ergänzen. Besonders bei körperlich anstrengenden Entladeaufgaben, wie z.B. der Gepäckentladung an Flughäfen, kann ein solches in-

teraktives Entladesystem eine erhebliche Entlastung des Menschen mit sich bringen.

2. Testsystem: Das interaktive Entladesystem kann dazu verwendet werden, neue Entladestrategien zu testen und zu evaluieren, ohne dass dazu aufwendig ein entsprechendes Programm erstellt werden muss. So können z.B. verschiedene Strategien getestet werden, bei denen die Reihenfolge, in der die Objekte entladen werden, variiert wird.
3. Benchmarksystem: Das interaktive Entladesystem kann als Benchmark für ein vollautomatisches Entladesystem dienen. Die Greifpositionen, die durch ein automatisches System ausgewählt werden, können mit manuell ausgewählten Greifpositionen in einem interaktiven Entladesystem verglichen werden. Ein Maßstab für die Qualität von Greifpositionen ist z.B. die Anzahl erfolgreicher Greifversuche.

In Abbildung 5.1 ist der vereinfachte Ablauf des semi-automatischen Modus in Form eines Ablaufdiagramms dargestellt. Zunächst wird der Behälter mit Inhalt, wie im vollautomatischen System auch, als 3D-Punktewolke erfasst. Anschließend wird der Nutzer aufgefordert, die nächste gewünschte Greifposition auszuwählen. Dies geschieht, je nach Eingabemodalität, entweder durch Markieren der betreffenden Position auf einem Bildschirm oder durch Verwendung eines speziellen Laserpointertools.

Nachdem eine Greifposition vorgegeben wurde, wird die entsprechende 3D-Position in der Punktewolke bestimmt und an der gewünschten Stelle ein optimaler Greifpunkt berechnet. Dazu wird wie im vollautomatischen System, ein Modell der Greiferkontaktfläche in die Punktewolke eingepasst (siehe Abschnitte 4.4 und 4.5).

Da die Einpassung des Greifermodells über ein lokales Optimierungsverfahren erfolgt, werden sich die gefundenen Lösungen im Allgemeinen in der Nähe der Vorgabeposition befinden.

Tritt der Fall ein, dass der Einpassungsfehler nach der Optimierung zu groß ist, oder dass die optimierte Greifposition zu weit von der vorgegebenen Position entfernt ist, so wird dies dem Nutzer mitgeteilt und er wird aufgefordert, eine neue Greifposition vorzugeben. Falls ein gültiger Greifpunkt gefunden wurde, wird versucht, eine kollisionsfreie Bahn zur Greifposition zu berechnen. Falls eine solche existiert, kann der Greifvorgang gestartet werden. Die Teile des Ablaufs, die die Bahnberechnung und die Ausführung der Roboterbewegung beinhalten, unterscheiden sich nicht von denen des vollautomatischen Systems. Nach der erfolgreichen Objektentnahme beginnt der Prozesszyklus von Neuem, solange bis

der Behälter entleert worden ist.

5.2 Bildschirmbasiertes semi-automatisches Entladesystem

5.2.1 Eingabe über einen Bildschirm

Bei der Vorgabe des Greifpunktes per Bildschirmeingabe, wird dem Benutzer ein Kamerabild des Behälters mit Inhalt angezeigt. Der Benutzer muss den gewünschten Greifpunkt in dem Bild markieren, z.B. durch Anklicken mit der Maus. Denkbar ist aber auch die Verwendung eines mobilen Touchscreens, bei dem die Auswahl über Berührung erfolgt.

Da die Position und die Orientierung der Kamera durch die Kalibrierung bekannt sind, ist es möglich, einen virtuellen Sichtstrahl zu berechnen. Dieser Strahl beginnt an der Position der Kamera und verläuft durch die markierte Position in der Bildebene.

Seien (u, v) die Pixelkoordinaten der angeklickten Position im Bild. Dann gilt für die Richtung \vec{n} des Sichtstrahls ausgedrückt in Weltkoordinaten:

$$\vec{n} = R_{WK} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \quad (5.1)$$

mit

$$x = \frac{u - c_x}{f_x} \quad \text{und} \quad y = \frac{v - c_y}{f_y} \quad (5.2)$$

R_{WK} ist die 3×3 -Matrix, welche die Rotation zwischen Welt- und Kamerakoordinatensystem beschreibt und c_x, c_y, f_x, f_y sind die intrinsischen Kameraparameter aus Abschnitt 4.1.1. Die Parameter zur Beschreibung der Linsenverzerrung sind hier nicht aufgeführt, da wir von einem bereits entzerrten Bild ausgehen.

Der Strahl kann mit der zuvor aufgenommenen Punktwolke geschnitten werden (siehe Abb. 5.2). Falls ein Schnittpunkt gefunden wurde, bildet dieser Punkt die initiale Position für die Greifpunktbestimmung.

5.2.2 Berechnung des Sichtstrahlschnittpunktes

Um den Schnittpunkt des Sichtstrahls mit der 3D-Punktwolke zu ermitteln, wird der Sichtstrahl als Zylinder mit einem Durchmesser d und einer Länge l aufgefasst

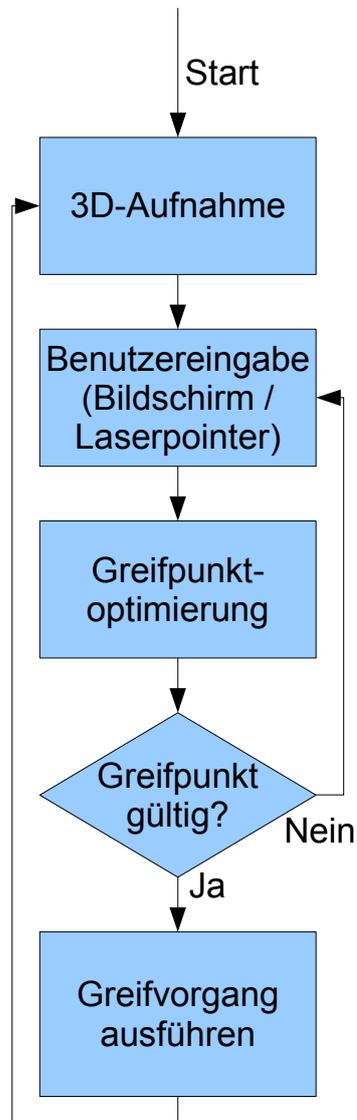


Abbildung 5.1: Vereinfachtes Ablaufdiagramm für den semi-automatischen Modus.

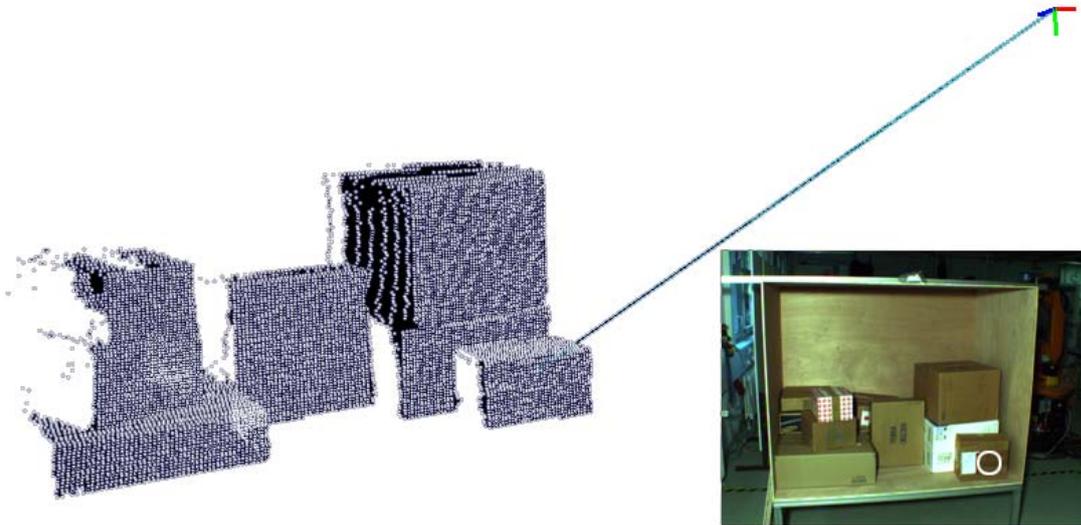


Abbildung 5.2: Das Koordinatenkreuz zeigt Position und Orientierung der Kamera an. Nachdem im 2D-Bild eine Position markiert worden ist (weißer Kreis), kann ein virtueller Sichtstrahl berechnet werden, der mit der Punktwolke geschnitten wird.

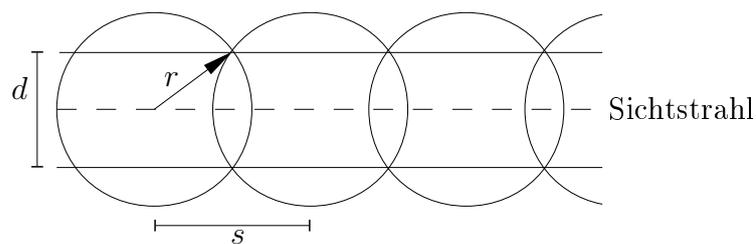


Abbildung 5.3: Der Sichtstrahl wird für die Schnittpunktberechnung durch Kugeln approximiert.

und durch eine Aneinanderreihung sich überlappender Kugeln approximiert (siehe Abb. 5.3).

Vorgegeben wird der Zylinderdurchmesser d und eine Schrittweite der Kugeln s . Um den Zylinder vollständig durch Kugeln abzudecken, muss für den Radius r der Kugeln gelten:

$$r = \frac{\sqrt{d^2 + s^2}}{2} \quad (5.3)$$

Sei \vec{p}_0 die Position des Kameraursprungs und \vec{n} die normierte Richtung des Sichtstrahls, dann werden an äquidistanten Positionen

$$\vec{p}_i = \vec{p}_0 + i s \vec{n} \quad (5.4)$$

die jeweils nächsten Nachbarn innerhalb eines Radiuses r ermittelt. Die nächsten Nachbarn können dabei effizient mit Hilfe des kd-Trees berechnet werden. Falls ein nächster Nachbar q innerhalb des Radius existiert, wird seine Projektion

$$\vec{q}' = \vec{p}_0 + \lambda \vec{n}, \text{ mit } \lambda = (\vec{q} - \vec{p}_0) \vec{n} \quad (5.5)$$

auf den Sichtstrahl bestimmt. Gilt $\|\vec{q} - \vec{q}'\| < d$ wurde ein Schnittpunkt \vec{q}' gefunden. Falls sich in der Kugel mehrere nächste Nachbarn mit Abstand kleiner d befinden, wird als Schnittpunkt derjenige Projektionspunkt auf dem Sichtstrahl festgelegt, dessen Abstand zur Kamera am kleinsten ist.

5.2.3 Untersuchung der Genauigkeit der vorgegebenen Positionen

Aufgrund verschiedener Fehlerquellen, stimmen die durch den Benutzer vorgegebene Position und die berechnete Position nie ganz exakt überein. Um den Grad der Abweichung zu bestimmen, wurden in einer Versuchsanordnung zwanzig Marker im Bereich des zu entladenden Containers positioniert und vermessen (siehe Abbildung 5.4). Mit Hilfe eines am Roboter befestigten Messwerkzeuges konnten die Positionen der einzelnen Marker bezüglich des Weltkoordinatensystems bestimmt werden. Die ermittelten Daten bilden die sogenannten Ground-Truth-Daten (Referenzwerte), mit denen die berechneten Werte verglichen werden. Als Maß für die Abweichung wird der Winkel zwischen berechnetem Sichtstrahl und idealem Sichtstrahl genommen. Seien \vec{p}_0 die Kameraposition und \vec{n} die normierte Richtung des Sichtstrahls, der beim Anklicken eines Markers berechnet wurde. Sei weiterhin \vec{m} die vermessenen Position des angeklickten Markers, so ergibt sich der Abweichungswinkel

$$\alpha = \arccos \left(\frac{\vec{n}(\vec{m} - \vec{p}_0)}{\|\vec{m} - \vec{p}_0\|} \right) \quad (5.6)$$

Für die Versuchsreihe wurde jeder der zwanzig Marker im Kamerabild von Hand möglichst genau angeklickt und der entsprechende Winkel α berechnet. Es ergab sich ein durchschnittlicher Abweichungswinkel

$$\bar{\alpha} = 0.604^\circ \quad (5.7)$$

bei einer Standardabweichung von

$$\sigma(\alpha) = 0.448^\circ \quad (5.8)$$

Der Abweichungswinkel ergibt sich aus den akkumulierten Fehlern einer nicht-optimalen extrinsischen und intrinsischen Kalibrierung der Kamera, den nicht exakt angeklickten Markern, sowie Fehlern, die bei der Vermessung der Marker mit Hilfe des Roboters entstehen können.

Im untersuchten Anwendungsszenario befindet sich der zu entladende Container in einer Entfernung von ca. 2500mm zur Kamera. Der ermittelte Abweichungswinkel entspricht daher einer tatsächlichen Abweichung bezogen auf die Bildebene der Kamera von

$$\Delta p = 26.4\text{mm} = 2500\text{mm} \cdot \tan(0.604^\circ) \quad (5.9)$$

Bei der Versuchsreihe wurde darauf geachtet, die Marker beim Anklicken möglichst genau zu treffen. In einer realistischen Anwendung muss jedoch davon ausgegangen werden, dass die angeklickte und die eigentlich gewünschte Position im Kamerabild voneinander abweichen. Um einen Eindruck von der Größe dieses zusätzlichen Fehlers zu bekommen, ermitteln wir für unseren Systemaufbau diejenige Abweichung, die aus einer Anklickungenauigkeit von einem Bildpixel resultiert. Gehen wir wieder von einem Arbeitsabstand von 2500mm aus, einem gegebenen Öffnungswinkel der Kameraoptik von 56° und einer horizontalen, darstellbaren Auflösung von 640 Pixeln, so ergibt sich ein zusätzlicher Abweichungsfehler von

$$E_{\text{pixel}} = \frac{2 \cdot 2500\text{mm} \cdot \tan(56^\circ)}{640\text{Pixel}} = 4.2 \frac{\text{mm}}{\text{Pixel}} \quad (5.10)$$

Klicken wir beispielsweise im Durchschnitt zwei Pixel neben der gewünschten Bildposition, so erhalten wir einen durchschnittlichen Gesamtfehler von 34.8mm. Dieser Fehler ist für die gegebene Anwendung hinreichend klein.



Abbildung 5.4: Eingemessene Marker zur Untersuchung der Genauigkeit der angeklickten Positionen

5.3 Laserpointerbasiertes semi-automatisches Entladesystem

5.3.1 Laserpointer als Eingabegerät

Im Folgenden wird eine Mensch-Maschine-Schnittstelle auf Basis eines Laserpointertools vorgestellt [36]. Mit Hilfe des Lasers kann der Nutzer auf Objekte zeigen, welche vom Roboter gegriffen werden sollen. Das Zeigen mit einem Laserpointer ist sehr intuitiv, da diese Eingabemodalität einer natürlichen Zeigegeste sehr nahe kommt. Des weiteren ermöglicht die Verwendung eines solchen Eingabegerätes es dem Benutzer, sich relativ frei in einer Szene zu bewegen und auf Objekte zu zeigen, die ansonsten (aus der Sicht einer stationären Kamera) durch andere Objekte verdeckt wären.

Laserpointer werden für verschiedene Anwendungen im Bereich der Mensch-Maschine-Interaktion eingesetzt, z.B. für die Interaktion mit entfernten, großen Computerdisplays [56, 59] aber auch für die Steuerung von Robotern. Choi et al. [15] entwickelten einen mobilen Assistenzroboter, der beeinträchtigten Menschen beim Greifen und Heben alltäglicher Gegenstände helfen soll. Ein zu manipulie-

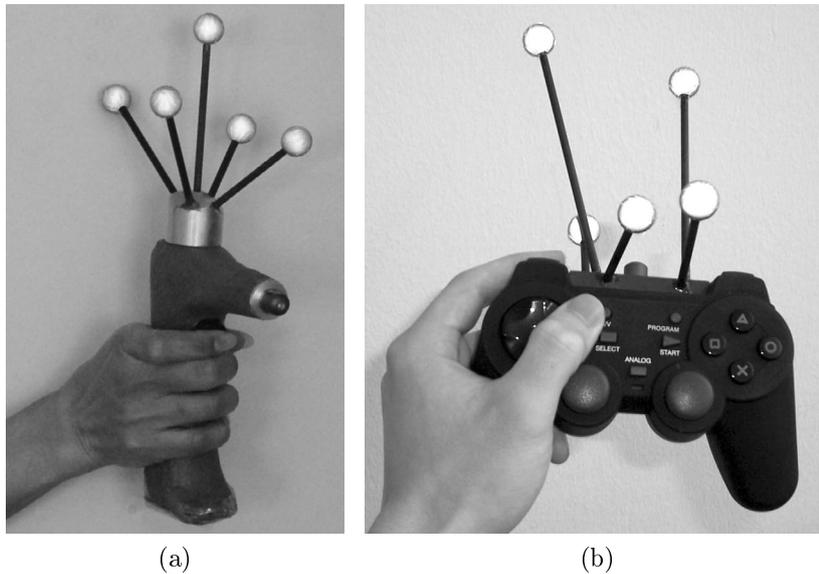


Abbildung 5.5: Laserpointertools. (a) IPR-Konstruktion mit Taster, (b) Umgebautes Gamepad mit zusätzlichen Eingabemöglichkeiten.

rendes Objekt wird hierbei mit einem Laserpointerinterface ausgewählt. Stopp et al. [76] benutzen einen Laserpointer, um eine Position auf einem Fließband anzuzeigen, an der ein Industrieroboter ein Objekt platzieren soll.

Die meisten existierenden Systeme dieser Art, benutzen Farbkameras, um den Laserpunkt zu detektieren. In unserem Ansatz dagegen wird das Laserpointertool mit trackbaren Markern ausgestattet, welche die Bestimmung der Position und der Orientierung des Eingabegerätes ermöglichen (Abbildung 5.5) und damit auch der Position und der Richtung des Laserstrahls. Der sichtbare Laserpunkt wird vom System nicht weiter ausgewertet, er dient ausschließlich als visuelles Feedback für den Benutzer. Durch diesen Ansatz ist das System unabhängig von den Reflexionseigenschaften der Objekte.

Im Folgenden wird ein Überblick über den Aufbau des laserpointerbasierten Entladesystems gegeben, anschließend die Kalibrierung des Laserpointertools beschrieben und danach einige experimentelle Ergebnisse präsentiert.

5.3.2 Systemüberblick

Abbildung 5.6 zeigt den prinzipiellen Aufbau eines möglichen laserpointerbasierten Entladesystems. Die Roboterzelle enthält einen 6-Achs-Industrieroboter, der

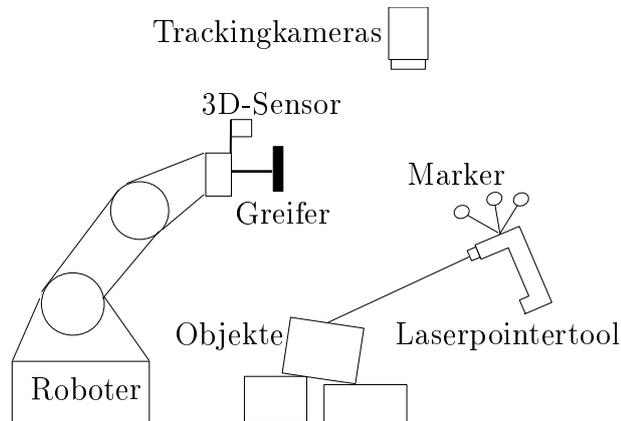


Abbildung 5.6: Systemaufbau des laserpointerbasierten Entladesystems.

je nach Anwendung mit einem Saug- oder Winkelgreifer ausgestattet ist.

Die Erfassung der Szene erfolgt in diesem Setup durch einen PMD-Sensor, welcher am Roboterflansch befestigt ist. Dadurch ist es möglich, mit dem Sensor um die Objekte herumzufahren und so eine vollständige 3D-Punktwolke aufzunehmen. Abbildung 5.7 zeigt eine 3D-Punktwolke, die durch Fusion mehrerer PMD-Einzelaufnahmen entstanden ist.

Die Vorgabe der Greifposition erfolgt mit dem Laserpointertool. Das Eingabegerät ist mit kugelförmigen Markern ausgestattet, die von einem Infrarot-Echtzeit-trackingsystem erkannt werden. Da die Anordnung der Marker dem Trackingssystem bekannt ist, kann es die Position und die Orientierung des Eingabetools eindeutig bestimmen.

Der Gesamtprozess ist analog zum bildschirmbasierten Entladesystem. Nach der 3D-Erfassung der Szene, wählt der Benutzer eine Greifposition aus, indem er das Laserpointertool auf die gewünschte Position richtet und die Eingabe durch Drücken eines Tasters bestätigt.

Danach wird die Position und die Richtung des Laserstrahls bestimmt. Nach dem Verfahren aus Abschnitt 5.2.2 kann ein Schnittpunkt zwischen dem Laserstrahl und der aufgenommenen 3D-Punktwolke berechnet werden, der als initiale Position für die Optimierung des Greifpunktes dient.

Die Abbildungen 5.8 und 5.9 zeigen zwei auf Laserpointertools basierende Entladesysteme. Das erste System beruht auf der Greifpunktbestimmung in einer 3D-Punktwolke (siehe Abschnitt 4.4). Hier werden mit Hilfe eines Vakuumgreifers Pakete gegriffen, die zuvor mit dem Laser markiert wurden. Das zweite System verwendet die Greifpunktbestimmung auf Tiefenbildern (siehe Abschnitt 4.5), um markierte Stofftiere mit Hilfe eines Winkelgreifers in eine Ablagebox zu befördern.

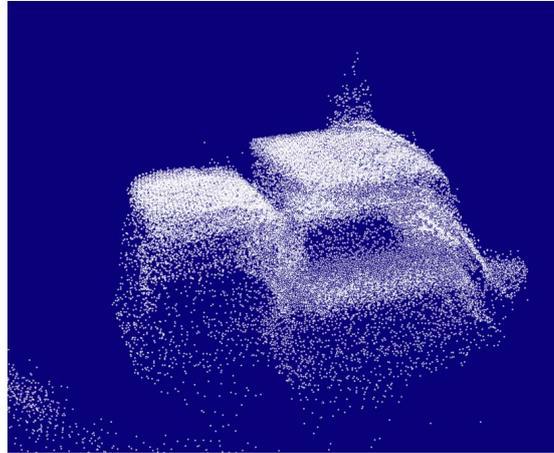


Abbildung 5.7: 3D-Punktwolke einer Szene mit Kartons, aufgenommen durch einen PMD-Sensor.

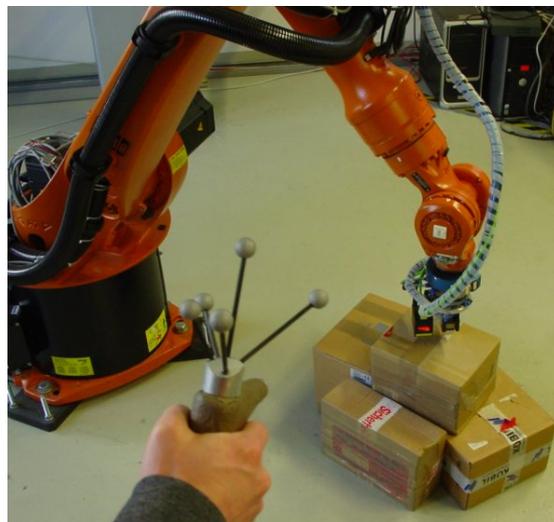


Abbildung 5.8: Der Nutzer markiert mit Hilfe des Laserpointertools den Karton, welcher als nächster gegriffen werden sollen.

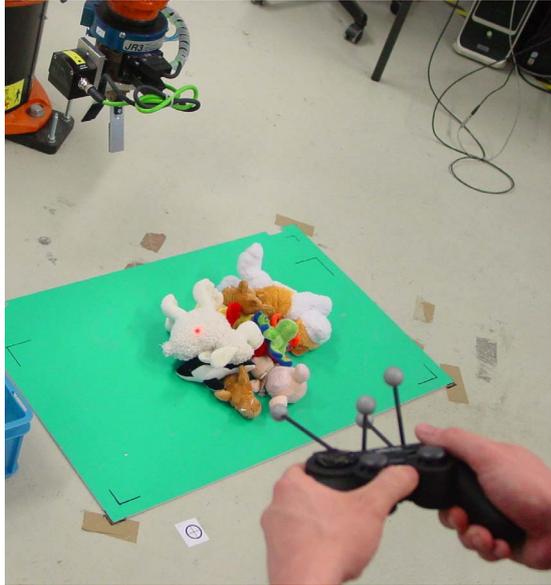


Abbildung 5.9: In dieser Applikation werden Stofftiere durch das Laserpointertool für den Greifvorgang ausgewählt.

5.3.3 Kalibrierung des Laserpointertools

Das Trackingsystem liefert die Position und die Orientierung des Markerkoordinatensystems. Da wir jedoch an der Position und der Richtung des Laserstrahls interessiert sind, müssen wir die Transformationsmatrix bestimmen, welche das Markerkoordinatensystem in das Laserpointerkoordinatensystem überführt. Wir haben dazu ein Kalibrierungsverfahren entwickelt, welches sich in zwei Schritte unterteilt.

Im ersten Schritt wird nur die Position der Lasertoolspitze (Laserdiode) geschätzt. Dazu wird die Spitze des Laserpointers um einen festen Referenzpunkt bewegt und die verschiedenen Positionen und Orientierungen des Markersystems gespeichert. Seien R_i und \vec{t}_i die Rotationsmatrizen bzw. die Translationsvektoren der gemessenen Markerframes. Dann erhalten wir die Position \vec{l}^m der Laserspitze, ausgedrückt in Markerkoordinaten und die Koordinaten des Referenzpunktes, ausgedrückt in Weltkoordinaten durch Minimierung von

$$E_{tip}(\vec{l}^m, \vec{p}^w) = \sum_i \left\| R_i \vec{l}^m + \vec{t}_i - \vec{p}^w \right\| \quad (5.11)$$

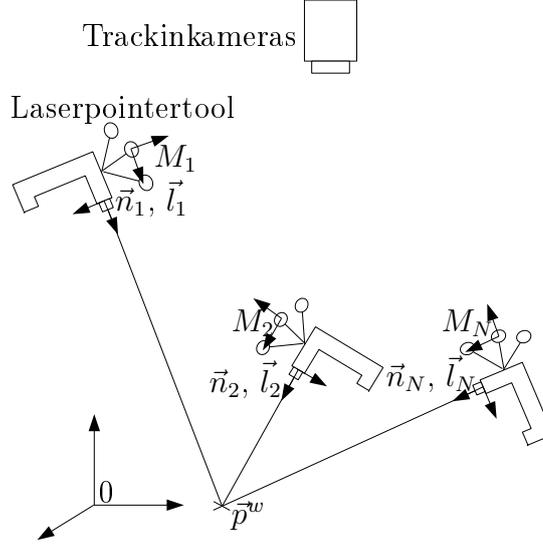


Abbildung 5.10: Für die Kalibrierung der Laserstrahlrichtung wird ein Referenzpunkt aus verschiedenen Richtungen anvisiert.

Die geschlossene Lösung dieses quadratischen Minimierungsproblems lautet:

$$\begin{pmatrix} \vec{l}^m \\ \vec{p}^w \end{pmatrix} = (A^T A)^{-1} A^T B \quad (5.12)$$

$$\text{mit } A = \begin{pmatrix} R_1, & -I_{3 \times 3} \\ R_2, & -I_{3 \times 3} \\ \vdots & \\ R_N, & -I_{3 \times 3} \end{pmatrix} \text{ und } B = \begin{pmatrix} -\vec{t}_1 \\ -\vec{t}_2 \\ \vdots \\ -\vec{t}_N \end{pmatrix}$$

Im nächsten Schritt wird die Richtung n des Laserstrahls berechnet. Dazu wird mit dem Laserpointer auf einen bekannten Referenzpunkt aus unterschiedlichen Richtungen und Positionen gezeigt (Abb. 5.10). Der benötigte bekannte Referenzpunkt \vec{p}^w kann z.B. aus dem ersten Kalibrierungsschritt übernommen werden oder mit Hilfe eines Roboters und einer Messspitze eingemessen werden.

Sei \vec{d}_i der Differenzvektor von der Laserpointerspitze zum Referenzpunkt ausgedrückt im Markerkoordinatensystem:

$$\vec{d}_i = R_i^{-1}(\vec{p}^w - \vec{t}_i) - \vec{l}^m \quad (5.13)$$

Dann können wir den jeweils kürzesten Abstand zwischen dem Referenzpunkt und den Laserstrahlen bestimmen. Die Summe dieser Abstände über alle Messungen

ergibt die Fehlerfunktion

$$\begin{aligned}
 E_{beam}(\vec{n}) &= \frac{1}{N} \sum_i \|\vec{d}_i - \vec{n}\vec{n}^T \vec{d}_i\|^2 \\
 &= \frac{1}{N} \sum_i (\vec{d}_i^T \vec{d}_i - \vec{d}_i^T \vec{n}\vec{n}^T \vec{d}_i) \\
 &= \frac{1}{N} \sum_i (\vec{d}_i^T \vec{d}_i - \vec{n}^T \vec{d}_i \vec{d}_i^T \vec{n}) \\
 &= -\vec{n}^T C \vec{n} + \text{const}
 \end{aligned} \tag{5.14}$$

$$\text{mit } C = \frac{1}{N} \sum_i \vec{d}_i \vec{d}_i^T \tag{5.15}$$

Um E_{beam} unter der Randbedingung $\vec{n}^T \vec{n} = 1$ zu minimieren, führen wir den Lagrangemultiplikator λ ein und erhalten:

$$\frac{\partial}{\partial \vec{n}} (\vec{n}^T C \vec{n} + \lambda(1 - \vec{n}^T \vec{n})) = 0 \tag{5.16}$$

Daraus ergibt sich:

$$C \vec{n} = \lambda \vec{n} \tag{5.17}$$

Setzen wir 5.17 in 5.14 ein, so erhalten wir schließlich

$$E_{beam}(\vec{n}) = -\vec{n}^T \lambda \vec{n} + \text{const} = -\lambda + \text{const} \tag{5.18}$$

Das bedeutet $E_{beam}(\vec{n})$ wird minimiert, wenn \vec{n} gleich dem Eigenvektor von C mit dem größten Eigenwert λ ist.

5.3.4 Experimentelle Ergebnisse

Nachdem die Kalibrierung des Laserpointers durchgeführt wurde, ergab sich für unseren Systemaufbau ein Fehler von

$$E_{beam}(n) = 224.1 \text{mm}^2 = (14.97 \text{mm})^2 \tag{5.19}$$

Das ist der mittlere quadratische Abstand zwischen dem Laserstrahl und dem Referenzpunkt. Der Fehler resultiert einerseits aus dem eigentlichen Kalibrierungsfehler, d.h. der Abweichung zwischen angenommenem und tatsächlichem Laserstrahl und andererseits aus unabsichtlichen Handbewegungen während der Kalibrierungsprozedur. Diese Zitterbewegungen, die sowohl beim Ruhighalten der

Hand als auch bei willkürlichen Bewegungen auftreten, werden als physiologischer oder normaler Tremor bezeichnet. Auch bei gesunden Menschen, lassen sich diese Bewegungen nicht unterdrücken.

Um zu Evaluieren, wie exakt eine gewünschte Zielposition mit dem Laserpointertool markiert werden kann, wurde eine Versuchsreihe mit mehreren Personen durchgeführt. Durch die Mittelung der Ergebnisse der Personen, kann der Einfluss eines besonders großen oder besonders kleinen individuellen Tremors minimiert werden.

Zunächst wurden insgesamt fünf Markierungen auf den Fußboden geklebt und mit Hilfe einer am Roboter befestigten Messspitze eingemessen. Dadurch war die exakte Position der Marker bekannt. Die Personen wurden anschließend angewiesen, mit dem Laserpointertool auf die Marker zu zielen und durch Drücken eines Tasters zu bestätigen, sobald der Marker nach ihrer Meinung hinreichend genau getroffen wurde.

Tabelle 5.1 stellt die Ergebnisse der Versuchsreihe dar. Bei einem mittleren Arbeitsabstand von ca. 1490mm ergab sich ein durchschnittlicher Abweichungswinkel α zwischen angenommenem und tatsächlichem Laserstrahl von 0.7690° bei einer Standardabweichung von $\sigma(\alpha) = 0.4285^\circ$. Daraus resultiert ein durchschnittlicher kürzester Abstand E_{Marker} zwischen dem Laserstrahl und den zu treffenden Markern von 19.7mm bei einer Standardabweichung von $\sigma(E_{Marker}) = 10.8\text{mm}$.

Für die gegebene Anwendung sind die ermittelten Genauigkeiten ausreichend, da die zu greifenden Objekte größer als der Fehler sind. Allerdings wird an dieser Stelle auch ein Nachteil des vorgestellten Ansatzes deutlich. Da die Position und die Orientierung des Laserstrahls am Tool selbst gemessen werden, wirken sich Messfehler des Trackingsystems sowie Fehler in der Kalibrierung umso stärker aus, je größer der Arbeitsabstand zwischen Tool und Objekten ist. Bei der direkten Detektion des Laserpunktes auf dem Objekt, tritt dieses Problem dagegen nicht auf.

	α [°]	$\sigma(\alpha)$ [°]	Abstand [mm]	E_{Marker} [mm]	$\sigma(E_{Marker})$ [mm]
Person 1	0.598	0.333	1513.5	15.643	8.431
Person 2	0.720	0.379	1339.1	16.765	8.686
Person 3	0.737	0.477	1459.5	18.999	13.663
Person 4	0.682	0.371	1698.0	19.904	10.534
Person 5	1.113	0.464	1418.9	27.246	9.936
Gesamt	0.769	0.429	1490.0	19.735	10.835

Tabelle 5.1: Evaluation der Genauigkeit des Laserpointertools. (α bezeichnet den mittleren Abweichungswinkel, $\sigma(\alpha)$ die Standardabweichung des Abweichungswinkels, mit Abstand ist die mittlere Distanz zwischen Laserpointertool und Markern gemeint, E_{Marker} bezeichnet die durchschnittliche kürzeste Distanz zwischen Laserstrahl und Markern und $\sigma(E_{Marker})$ ist die dazugehörige Standardabweichung).

6 Zusammenfassung und Ausblick

In diesem Kapitel werden die wissenschaftlichen Beiträge dieser Arbeit zusammengefasst und es werden mögliche Richtungen für weitere Forschungstätigkeiten aufgezeigt.

6.1 Beiträge der Arbeit

Im Rahmen dieser Arbeit wurden Verfahren der Bildverarbeitung für den Einsatz in einem generischen Entladesystem untersucht. Eine wichtige Randbedingung war, dass keine exakte Kenntnis über die Form und die Beschaffenheit der zu greifenden Objekte vorausgesetzt werden sollte. Es entstanden mehrere experimentelle Entladesysteme, in denen die Effektivität der vorgestellten Verfahren demonstriert werden konnte.

In der vorliegenden Arbeit wurden Beiträge in folgenden Bereichen erbracht:

Systemarchitektur Es wurde eine flexible Systemarchitektur für ein generisches Entladesystem konzipiert, die leicht an verschiedene Entladeaufgaben angepasst werden kann. Auf Basis einer vorhandenen Middleware wurde eine komponentenbasierte Softwarearchitektur realisiert. Die einzelnen Module lassen sich je nach Anforderungen neu zusammenschalten, so dass sie in unterschiedlichen Anwendungen wiederverwendet werden können. Durch die Kapselung der Sensoren in Sensorkomponenten und die Zurverfügungstellung einheitlicher Datenschnittstellen kann die 3D-Aufnahmetechnik ausgetauscht werden, ohne dass nachfolgende Verarbeitungseinheiten geändert werden müssen.

Kalibrierungsverfahren Für die verwendete 3D-Sensorik (Laserscanner, PMD-Sensor) wurden Verfahren zur Hand-Auge-Kalibrierung entwickelt. Die Ansätze zeichnen sich durch eine einfache Kalibrierungsprozedur aus, so dass sie auch im industriellen Umfeld leicht angewendet werden können, z.B. im Falle einer notwendigen Neu-Kalibrierung während des Betriebs.

Containerlokalisierung Bevor der Entladevorgang beginnen kann, muss die Position des Behälters bestimmt werden. Dies ist zum einen für eine kollisionsfreie Bahnplanung wichtig, zum anderen für die Greifpunktbestimmung, damit derjenige Bereich in der Szene bekannt ist, in dem sich zu greifende Objekte befinden. Für diese Aufgabe wurde ein modellbasiertes Verfahren realisiert, das die Containerposition und -orientierung anhand einer 3D-Aufnahme berechnet.

Greifpunktbestimmung Für die Greifpunktbestimmung wurde ein Verfahren entwickelt, bei dem ein Modell der Kontaktfläche des Greifers in die aufgenommenen 3D-Daten eingepasst wird. Dadurch können geeignete Greifpunkte gefunden werden, auch wenn kein Vorwissen über Form und Beschaffenheit der Objekte vorliegt, oder wenn sich die Objekte nur sehr schwer modellieren lassen. Es wurde ein Verfahren realisiert, welches Greifpunkte für einen Vakuumgreifer in einer 3D-Punktewolke bestimmt und ein weiteres, welches Greifpunkte für einen Zweifingergreifer in Tiefenbildern findet.

Semi-automatisches Entladesystem Zusätzlich zum vollautomatischen Entladesystem wurde ein semi-automatisches System realisiert. Ein Benutzer hat hierbei die Möglichkeit, das Entladesystem durch manuelle Vorgabe von Greifpunkten zu unterstützen. Ein solcher interaktiver Modus kann z.B. zum Einsatz kommen, wenn der vollautomatische Modus fehlschlägt, oder er kann zu Test- und Evaluationszwecken dienen. Zur Interaktion mit dem System wurden zwei verschiedene Mensch-Roboter-Schnittstellen entwickelt. Zum einen können Greifpunkte über eine bildschirmbasierte GUI markiert werden, zum anderen können gewünschte Greifpositionen über ein neuartiges Laserpointertool direkt in der Szene ausgewählt werden. Für beide Ansätze wurde eine Methode entwickelt, bei der ein virtueller Sichtstrahl mit einem zuvor aufgenommenem Umweltmodell geschnitten wird. Das Laserpointertool muss vor Benutzung einmal kalibriert werden. Hierfür wurde ein entsprechendes Kalibrierverfahren entwickelt.

6.2 Ausblick

Eine große Herausforderung bei der Lösung des „Griff in die Kiste“-Problems ist die Bestimmung geeigneter Greifpositionen anhand der Sensordaten, so dass ein Roboter die Objekte sicher greifen und ohne Beschädigungen aufnehmen kann. Insbesondere wenn die zu greifenden Objekte unbekannt sind, beliebig angeordnet und möglicherweise verklemmt sind, stellt dies ein äußerst komplexes Problem dar.

In der vorliegenden Arbeit konnte gezeigt werden, dass auf Grundlage von 3D-Daten sinnvolle Greifpunkte für unbekannte Objekten berechnet werden können. Die untersuchten Entladeaufgaben sind jedoch verhältnismäßig einfach, da ausschließlich Vakuumgreifer und Zweifingergreifer zum Einsatz kamen, die nur eine ebene Kontaktfläche bzw. zwei gegenüberliegende Flächen für einen sicheren Griff benötigen. Die Greifpunktbestimmung konnte einfach und flexibel gehalten werden, da auf eine Objekterkennung verzichtet wurde. Das hat jedoch den Nachteil, dass die gefundenen Greifpunkte im Allgemeinen nicht optimal sind. So setzt der Greifer meistens nicht im Schwerpunkt des Objektes an, oder der Greifer erfasst u.U. zwei nebeneinander liegende Objekte.

Zur Lösung des allgemeinen „Griff in die Kiste“-Problems müssen robuste Verfahren zur Objektsegmentierung gefunden werden. Nur wenn Objektgrenzen bekannt sind, können geeignete Greifpunkte auch für universelle Mehrfingergreifer gefunden werden, kann die Lage der Objekte untereinander analysiert und auf gegenseitige Verklemmungen hin untersucht werden. Existiert kein oder nur wenig Vorwissen über die zu greifenden Objekte, ist es sinnvoll, verschiedene Sensordaten, z.B. 3D-Daten und 2D-Farbbilder, miteinander zu fusionieren, um die Robustheit der Segmentierung zu erhöhen.

Glossar

CAD	Computer Aided Design
GENESYS	Generisches Entladesystem - BmBF-gefördertes Forschungsprojekt
GUI	Graphical User Interface
HMI	Human Machine Interface
HRI	Human Robot Interface
ICP	Iterative-Closest-Point-Algorithmus
IPO-Takt	Interpolationstakt einer Robotersteuerung
PMD	Photonic Mixer Device - 3D-Sensor
RANSAC	Random Sample Consensus
KMU	Kleine und mittlere Unternehmen
TCP	Tool Center Point
TOF	Time of Flight

Abbildungsverzeichnis

2.1	Entnahmesystem von Müller et al.	8
2.2	Prototyp eines Paketentladeroboters	9
2.3	Entnahmesystem nach Stotz et al.	10
2.4	Roboterassistenzsystem	14
2.5	Laserpointer als Eingabegerät	15
2.6	Trackbares Eingabegerät	16
3.1	Entladezelle	24
3.2	Hardwarearchitektur	24
3.3	Softwarearchitektur	26
3.4	Virtuelle Roboterzelle	29
3.5	Boundingboxen	29
3.6	Ablaufdiagramm	31
3.7	Entladesequenz	32
4.1	Schachbrett-Kalibriermuster	36
4.2	Kalibrierung der Kamera	37
4.3	Kalibrierung des Laserscanners	39
4.4	Kalibrierplatte und Laserscanner	40
4.5	Punktewolke der Kalibrierplatte	40
4.6	Kalibrierkörper für den Laserscanner	41
4.7	Kalibrierung des PMD-Sensors	43
4.8	Containermodell	49
4.9	Containerlokalisierung	49
4.10	Clipping der 3D-Punktewolke	50
4.11	Der ICP-Algorithmus als Pseudocode	53
4.12	Modellierung der Greiferkontaktflächen	55
4.13	Einpassung des Greiferflächenmodells	56
4.14	Ergebnis der Greifpunktbestimmung	56
4.15	Greifpunktbestimmung als Pseudocode	57
4.16	Häufigkeitsverteilung der Einpassungsfehler	58
4.17	Häufigkeitsverteilung der Zeiten	59

4.18	Tiefenbildgenerierung als Pseudocode	60
4.19	Berechnung des Tiefenbildes	62
4.20	Berechnung von Greifpunkten in einem Tiefenbild	63
4.21	Greifpunktbestimmung in Tiefenbildern	65
4.22	Nächste Nachbarn aus 2D-Punktemenge als Pseudocode	66
4.23	Greifen von flexiblen Objekten	67
5.1	Ablaufdiagramm für den semi-automatischen Modus	72
5.2	Greifpunktvorgabe über Bildschirm	73
5.3	Sichtstrahlapproximation	73
5.4	Marker zur Untersuchung der Anklickgenauigkeit	76
5.5	Laserpointertool	77
5.6	Laserpointerbasiertes Entladesystem	78
5.7	3D-Punktwolke	79
5.8	Vorgabe von Greifpunkten über das Laserpointertool	79
5.9	Markierung von Stofftieren durch Laserpointertool	80
5.10	Kalibrierung des Laserpointertools	81

Tabellenverzeichnis

2.1	Vergleich verschiedener Entladesysteme	11
4.1	Ergebnisse der Laserscanner-Kalibrierung mit Einfachscan.	42
4.2	Ergebnisse der Laserscanner-Kalibrierung mit Mehrfachscan.	43
4.3	Ergebnisse der Hand-Auge-Kalibrierung	47
4.4	Ergebnisse der Hand-Auge-Kalibrierung	48
4.5	Ergebnisse der Containerlokalisierung	51
5.1	Evaluation der Genauigkeit des Laserpointertools	84

Literaturverzeichnis

- [1] *Das generische Entladesystem.* <http://genesys.ira.uka.de>, . – letzter Zugriff 29.09.2009
- [2] ARUN, K. ; HUANG, T. ; BLOSTEIN, S.: Least-squares fitting of two 3-D point sets. In: *IEEE Transactions on pattern analysis and machine intelligence* 9 (1987), Nr. 5, S. 698–700
- [3] BALSLEV, I. ; ERIKSEN, R. D.: From belt picking to bin packing. In: *Proceedings of SPIE - The International Society for Optical Engineering* Bd. 4902, 2002, S. 616–623
- [4] BENTLEY, J. L.: Multidimensional divide-and-conquer. In: *Communications of the ACM* 23 (1980), Nr. 4, S. 214–229
- [5] BERGER, M. ; BACHLER, G. ; SCHERER, S.: Vision guided bin picking and mounting in a flexible assembly cell. In: *Proceedings of the 13th international conference on Industrial and engineering applications of artificial intelligence and expert systems*, 2000, S. 109–118
- [6] BESL, P. ; MCKAY, H.: A method for registration of 3-D shapes. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 14 (1992), Feb, Nr. 2, S. 239–256
- [7] BICCHI, A. ; KUMAR, V.: Robotic grasping and contact: a review. In: *Proceedings of the 2000 IEEE International Conference on Robotics and Automation* Bd. 1, 2000, S. 348–353
- [8] BOEHNKE, K.: Object localization in range data for robotic bin picking. In: *Proceedings of the 2007 IEEE International Conference on Automation Science and Engineering*, 2007, S. 572–577
- [9] BOUGHORBEL, F. ; ZHANG, Y. ; KANG, S. ; CHIDAMBARAM, U. ; ABIDI, B. ; KOSCHAN, A. ; ABIDI, M.: Laser ranging and video imaging for bin picking. In: *Assembly Automation* 23 (2003), Nr. 1, S. 53–59

- [10] BROWN, D.: Close-range camera calibration. In: *Photogrammetric engineering* 37 (1971), Nr. 8, S. 855–866
- [11] BRUYNINCKX, H.: Open robot control software: the OROCOS project. In: *Proceedings of the 2001 IEEE International Conference on Robotics and Automation* Bd. 3, 2001, S. 2523–2528
- [12] BRUYNINCKX, H. ; SOETENS, P. ; KONINCKX, B.: The real-time motion control core of the Orocos project. In: *Proceedings of the 2003 IEEE International Conference on Robotics and Automation* Bd. 2, 2003, S. 2766–2771
- [13] CANNY, J.: A Computational Approach to Edge Detection. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 8 (1986), Nr. 6, S. 679–698
- [14] CHEN, Y. ; MEDIONI, G.: Object modeling by registration of multiple range images. In: *Proceedings of the 1991 IEEE International Conference on Robotics and Automation* Bd. 3, 1991, S. 2724–2729
- [15] CHOI, Y. S. ; ANDERSON, C. D. ; GLASS, J. D. ; KEMP, C. C.: Laser pointers and a touch screen: intuitive interfaces for autonomous mobile manipulation for the motor impaired. In: *Proceedings of the 10th international ACM SIGACCESS conference on Computers and accessibility*, 2008, S. 225–232
- [16] CHRISTOPOULOS, V. ; SCHRATER, P.: Handling shape and contact location uncertainty in grasping two-dimensional planar objects. In: *Proceedings of the 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2007, S. 1557–1563
- [17] CRAIG, J. J.: *Introduction to Robotics: Mechanics and Control*. 3. Prentice Hall, 2004. – ISBN 13 978–0201543612
- [18] CREMERS, D.: *Statistical shape knowledge in variational image segmentation*, Department of Mathematics and Computer Science, University of Mannheim, Germany, Diss., 2002
- [19] DING, D. ; LIU, Y. ; WANG, S.: Computing 3-D optimal form-closure grasps. In: *Proceedings of the 2000 IEEE International Conference on Robotics and Automation* Bd. 4, 2000, S. 3573 – 3578

- [20] ECHELMAYER, W. ; TANK, S. ; WELLBROCK, E.: Einsatzmöglichkeiten von Industrierobotern in Paketverteilzentren. In: *Fraunhofer Institut für Produktionstechnik und Automatisierung: Roboter in der Intralogistik* (2005), S. 78–86
- [21] EGGERT, D. ; LORUSSO, A. ; FISHER, R.: Estimating 3-D rigid body transformations: a comparison of four major algorithms. In: *Machine Vision and Applications* 9 (1997), Nr. 5, S. 272–290
- [22] FISCHLER, M. A. ; BOLLES, R. C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. In: *Communications of the ACM* 24 (1981), Nr. 6, S. 381–395
- [23] GOCKEL, T. ; AHLMANN, J. ; DILLMANN, R. ; AZAD, P.: 3D Vision Sensing for Grasp Planning: A New, Robust and Affordable Structured Light Approach. In: *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, 2005, S. 2529–2534
- [24] GRANLUND, G. ; KNUTSSON, H.: *Signal processing for computer vision*. MA, USA : Kluwer Academic Publishers Norwell, 1995. – ISBN 13 978–0792395300
- [25] GUTHART, G. ; SALISBURY, J.: The Intuitive TM telesurgery system: Overview and application. In: *Proceedings of the IEEE International Conference on Robotics and Automation* Bd. 1, 2000, S. 618–621
- [26] HELMS, E. ; SCHRAFT, R. D. ; HAGELE, M.: rob@work: Robot assistant in industrial environments. In: *Proceedings of the 11th IEEE International Workshop on Robot and Human Interactive Communication*, 2002, S. 399–404
- [27] HENRICH, D. ; CHENG, X. ; REMBOLD, U. ; DILLMANN, R.: Fast Distance Computation for on-line Collision detection with multi-arm robots. In: *Proceedings of the IEEE International Conference on Robotics and Automation* Bd. 3, 1992, S. 2514–2519
- [28] HESSE, S.: *Grundlagen der Handhabungstechnik*. Carl Hanser Verlag, 2006
- [29] HORAUD, R. ; DORNAIKA, F.: Hand-eye calibration. In: *The international journal of robotics research* 14 (1995), Nr. 3, S. 195
- [30] HORN, B. ; IKEUCHI, K.: The Mechanical Manipulation of Randomly Oriented Parts. In: *Scientific American* 251 (1984), Nr. 2, S. 100 – 109

- [31] IKEUCHI, K. ; HORN, B. K. P. ; NAGATA, S. ; CALLAHAN, T. ; FEINGOLD, O.: Picking up an object from a pile of objects. In: *Proceedings of the First International Symposium on Robotics Research*, MIT Press, 1983, S. 139–166
- [32] IKONEN, L. ; KÄLVIÄINEN, H. ; OINONEN, O.: A computer vision approach for robotized handling of sheets in a manufacturing cell. In: *Proceedings of the Scandinavian conference in image analysis 1* (1997), S. 309–316
- [33] KAISER, B. ; TAURO, R. A. ; HEIN, B. ; WÖRN, H.: Roboterbasierte Halbautomatische Entladung. In: *VDI-Berichte 2012, Robotik 2008*, 2008, S. 139–142
- [34] KAISER, B. ; TAURO, R. A. ; WÖRN, H.: Extrinsic calibration of a robot mounted 3D imaging sensor. In: *International Journal of Intelligent Systems Technologies and Applications* 5 (2008), Nr. 3-4, S. 374–379
- [35] KAISER, B. ; TAURO, R. A. ; WÖRN, H.: Automatic and Semi-automatic Unloading of Containers. In: *Robotics, Telematics and Applications*, 2009. – akzeptiert
- [36] KAISER, B. ; TAURO, R. A. ; WÖRN, H.: Laser Pointing to Control a Roboter. In: *Robotics, Telematics and Applications*, 2009. – akzeptiert
- [37] KATSOULAS, D.: Reliable recovery of piled box-like objects via parabolically deformable superquadrics. In: *Proceedings of the Ninth IEEE International Conference on Computer Vision* Bd. 2, 2003, S. 931–938
- [38] KATSOULAS, D. ; BERGEN, L. ; TASSAKOS, L.: A versatile depalletizer of boxes based on range imagery. In: *Proceedings of the 2002 IEEE International Conference on Robotics and Automation* Bd. 4, 2002, S. 4313–4319
- [39] KATSOULAS, D.: *Robust recovery of piled box-like objects in range images*, Universität Freiburg, Diss., 2005
- [40] KAVOUSSANOS, M. ; POULIEZOS, A.: Visionary automation of sack handling and emptying. In: *IEEE Robotics & Automation Magazine* 7 (2000), Dec, Nr. 4, S. 44–49
- [41] KEMP, C. C. ; ANDERSON, C. D. ; NGUYEN, H. ; TREVOR, A. J. ; XU, Z.: A point-and-click interface for the real world: laser designation of objects for mobile manipulation. In: *Proceedings of the 3rd ACM/IEEE international conference on Human robot interaction*. New York, USA, 2008, S. 241–248

-
- [42] KIRKEGAARD, J. ; MOESLUND, T.: Bin-Picking based on Harmonic Shape Contexts and Graph-Based Matching. In: *Proceedings of the 18th International Conference on Pattern Recognition* Bd. 2, 2006, S. 581–584
- [43] KRAISS, K.: *Advanced man-machine interaction: fundamentals and implementation*. Springer Verlag, 2006
- [44] KRISTENSEN, S. ; ESTABLE, S. ; KOSSOW, M. ; BRÖSEL, R.: Bin-picking with a solid state range camera. In: *Robotics and Autonomous Systems* 35 (2001), Nr. 3-4, 143–151
- [45] LAVALLE, S. M.: Rapidly-exploring random trees: A new tool for path planning / Department of Computer Science Iowa State University Ames. 1998. – Forschungsbericht
- [46] LINDNER, M. ; KOLB, A.: Lateral and Depth Calibration of PMD-Distance Sensors. In: *Advances in Visual Computing* Bd. 2, Springer, 2006, S. 524–533
- [47] LOWE, D. G.: Object Recognition from Local Scale-Invariant Features. In: *Proceedings of the IEEE International Conference on Computer Vision* Bd. 2, 1999, S. 1150
- [48] MAYER, H. ; NAGY, I. ; KNOLL, A. ; BRAUN, E. ; BAUERNSCHMITT, R. ; LANGE, R.: Haptic feedback in a telepresence system for endoscopic heart surgery. In: *Presence: Teleoperators and Virtual Environments* 16 (2007), Nr. 5, S. 459–470
- [49] MEYER, C. ; HOLLMANN, R. ; PARLITZ, C. ; HÄGELE, M.: Programmieren durch Vormachen für Assistenzsysteme - Schweiss- und Klebebahnen intuitiv programmieren. In: *it-Information Technology* 49 (2007), Nr. 4, S. 238–246
- [50] MIAN, A. S. ; BENNAMOUN, M. ; OWENS, R.: Three-Dimensional Model-Based Object Recognition and Segmentation in Cluttered Scenes. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 28 (2006), Nr. 10, S. 1584–1601
- [51] MILLER, A. ; ALLEN, P.: GraspIt! a versatile simulator for robotic grasping. In: *IEEE Robotics & Automation Magazine* 11 (2004), Nr. 4, S. 110–122
- [52] MILLER, A. ; KNOOP, S. ; CHRISTENSEN, H. ; ALLEN, P.: Automatic grasp planning using shape primitives. In: *Proceedings of the 2003 IEEE International Conference on Robotics and Automation* Bd. 2, 2003, S. 1824–1829

- [53] MÜLLER, M. ; WÖRN, H.: A new generation robot system for object picking. In: *International Conference on Multisensor Fusion and Integration for Intelligent Systems*, 2001, S. 55–60
- [54] MÜLLER, M.: *Entnahmestrategien zur Objektvereinzelung mittels Industrierobotern*, Universität Karlsruhe (TH), Diss., 2001
- [55] MÜLLER, M. ; WÖRN, H.: Bin-Picking Systems and Applications - A State of the Art. In: *Proceedings of the IASTED International Conference Robotics and Automation RA*, 2000, S. 1–10
- [56] MYERS, B. A. ; BHATNAGAR, R. ; NICHOLS, J. ; PECK, C. H. ; KONG, D. ; MILLER, R. ; LONG, A. C.: Interacting at a distance: measuring the performance of laser pointers and other devices. In: *Proceedings of the 2002 SIGCHI conference on Human factors in computing systems*, 2002, S. 33–40
- [57] NGUYEN, V.-D.: Constructing stable force-closure grasps. In: *Proceedings of 1986 ACM Fall joint computer conference*. Los Alamitos, CA, USA : IEEE Computer Society Press, 1986, S. 129–137
- [58] NISSUM, K. ; LARSEN, T. ; MADSEN, O. ; NIELSEN, H.: Virtual 3D Environment for Planning Robotic Paint Routes. In: *Proceedings of the 2007 IEEE International Conference on Robotics and Automation*,, 2007, S. 4776–4781
- [59] OLSEN, D. R. Jr. ; NIELSEN, T.: Laser pointer interaction. In: *Proceedings of the SIGCHI conference on Human factors in computing systems*. New York, NY, USA : ACM, 2001, S. 17–22
- [60] PIRES, J. ; GODINHO, T. ; NILSSON, K. ; HAAGE, M. ; MEYER, C.: Programming industrial robots using advanced input-output devices: test-case example using a CAD package and a digital pen based on the Anoto technology. In: *International Journal of Online Engineering (iJOE)* 3 (2007), Nr. 3
- [61] PONCE, J. ; STAM, D. ; FAVERJON, B.: On computing two-finger force-closure grasps of curved 2D objects. In: *The International Journal of Robotics Research* 12 (1993), Nr. 3, S. 263
- [62] PRESS, W. H. ; TAUKOLSKY, S. A. ; VETTERLING, W. T. ; FLANNERY, B. P.: *Numerical recipes*. New York, USA : Cambridge University Press, 2002. – ISBN 13 978–0–521–75033–2

- [63] RAHARDJA, K. ; KOSAKA, A.: Vision-based bin-picking: recognition and localization of multiple complex objects using simple visual cues. In: *Proceedings of the 1996 IEEE/RSJ International Conference on Intelligent Robots and Systems* Bd. 3, 1996, S. 1448–1457
- [64] ROGALLA, O. ; EHRENMANN, M. ; ZOLLNER, R. ; BECHER, R. ; DILLMANN, R.: Using gesture and speech control for commanding a robot assistant. In: *Proceedings of the 11th IEEE International Workshop on Robot and Human Interactive Communication*, 2002, S. 454–459
- [65] RUSINKIEWICZ, S. ; LEVOY, M.: Efficient Variants of the ICP Algorithm. In: *Proceedings of the Third International Conference on 3-D Digital Imaging and Modeling*, 2001, S. 145–152
- [66] SAKAMOTO, D. ; HONDA, K. ; INAMI, M. ; IGARASHI, T.: Sketch and run: a stroke-based interface for home robots. In: *Proceedings of the 27th international conference on Human factors in computing systems* ACM New York, NY, USA, 2009, S. 197–200
- [67] SALONIA, M. ; HEIN, B. ; WÖRN, H.: Fast approximated conversion of workspace distances into free regions in configuration space of robots. In: *Proceedings of 37th International Symposium on Robotics and the 4th German Conference on Robotics*, VDI Verlag, 2006
- [68] SAXENA, A. ; DRIEMEYER, J. ; NG, A. Y.: Robotic Grasping of Novel Objects using Vision. In: *The International Journal of Robotics Research* 27 (2008), Nr. 2, 157-173
- [69] SCHMID, A. J. ; HOFFMANN, M. ; WÖRN, H.: A tactile language for intuitive human-robot communication. In: *Proceedings of the 9th international conference on Multimodal interfaces*. New York, NY, USA : ACM, 2007, S. 58–65
- [70] SCHOLTZ, J. ; YOUNG, J. ; DRURY, J. ; YANCO, H.: Evaluation of human-robot interaction awareness in search and rescue. In: *Proceedings of the 2004 IEEE International Conference on Robotics and Automation* Bd. 3, 2004, S. 2327–2332
- [71] SCHOLZ-REITER, B. ; ECHELMMEYER, I. ; WELLBROCK, D.: Development of a Robot-Based System for Automated Unloading of Variable Packages out of Transport Units and Containers. In: *Proceedings of the 2008 IEEE International Conference on Automation and Logistics*, 2008, S. 2766–2770

- [72] SCHRAFT, R. ; LEDERMANN, T.: Intelligent picking of chaotically stored objects. In: *Assembly Automation* 23 (2003), Nr. 1, S. 38–42
- [73] SCHRAFT, R. ; MEYER, C. ; PARLITZ, C. ; HELMS, E.: PowerMate - A Safe and Intuitive Robot Assistant for Handling and Assembly Tasks. In: *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, 2005, S. 4074–4079
- [74] SHIU, Y. ; AHMAD, S.: Calibration of wrist-mounted robotic sensors by solving homogeneous transform equations of the form $AX=XB$. In: *IEEE Transactions on Robotics and Automation* 5 (1989), Nr. 1, S. 16–29
- [75] STEINFELD, A. ; FONG, T. ; KABER, D. ; LEWIS, M. ; SCHOLTZ, J. ; SCHULTZ, A. ; GOODRICH, M.: Common metrics for human-robot interaction. In: *Proceedings of the 1st ACM SIGCHI/SIGART conference on Human-robot interaction* ACM New York, NY, USA, 2006, S. 33–40
- [76] STOPP, A. ; BALDAUF, T. ; HANTSCH, R. ; HORSTMANN, S. ; KRISTENSEN, S. ; LOHNERT, F. ; PRIEM, C. ; RUSCHER, B.: The Manufacturing Assistant: safe, interactive teaching of operation sequences. In: *Proceedings of the 11th IEEE International Workshop on Robot and Human Interactive Communication*, 2002, S. 386–391
- [77] STOPP, A. ; HORSTMANN, S. ; KRISTENSEN, S. ; LOHNERT, F.: Toward interactive learning for manufacturing assistants. In: *IEEE Transactions on Industrial Electronics* 50 (2003), Nr. 4, S. 705–707
- [78] STOTZ, M. ; VERL, A.: Ein neuer Ansatz zur Erkennung der Objektlage von ungeordneten Bauteilen für die automatisierte Handhabung. In: *VDI-Berichte 2012, Robotik 2008*, 2008, S. 131–134
- [79] TANK, S. ; ECHELMAYER, W.: Der Paketroboter. Optimierung logistischer Prozesse durch eine Weltneuheit. In: *Industrie Management* 22 (2006), Nr. 2, S. 13–16
- [80] TAYLOR, R. ; STOIANOVICI, D.: Medical robotics in computer-integrated surgery. In: *IEEE Transactions on Robotics and Automation* 19 (2003), Nr. 5, S. 765–781
- [81] TSAI, R. Y. ; LENZ, R. K.: A New Technique for Fully Autonomous and A New Technique for Fully Autonomous and Efficient 3D Robotics Hand/Eye

- Calibration. In: *IEEE Transactions on Robotics and Automation* 5 (1989), Nr. 3, S. 345–358
- [82] WOLF, A. ; STEINMANN, R. ; SCHUNK, H.: *Grippers in motion: the fascination of automated handling tasks*. Springer, 2005
- [83] XU, Z. ; SCHWARTE, R. ; HEINOL, H. ; BUXBAUM, B. ; RINGBECK, T.: Smart pixel - photonic mixer device (PMD) / PMD Technologies. 1999. – Forschungsbericht
- [84] YANCO, H. ; DRURY, J. ; SCHOLTZ, J.: Beyond usability evaluation: Analysis of human-robot interaction at a major robotics competition. In: *Human-Computer Interaction* 19 (2004), Nr. 1, S. 117–149
- [85] ZHANG, Q. ; PLESS, R.: Extrinsic Calibration of a Camera and Laser Range Finder. In: *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*, 2004, S. 2301–2306
- [86] ZHANG, Z.: A flexible new technique for camera calibration. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22 (2000), Nr. 11, S. 1330–1334