



Universität Hamburg

# 3D Raumdarstellung und Rekonstruktion geometrischer Primitive mittels eines 2D Laserscanners

DIPLOMARBEIT

im Studiengang Informatik

Diplomandin: Mehtap Arli  
Department: Informatik  
Arbeitsbereich: TAMS - Technische Aspekte Multimodaler Systeme  
Betreuer: Prof. Dr. Jianwei Zhang  
Zweitkorrektor: Dr. Werner Hansmann  
Abgabedatum: 18. Februar 2010



### **Zusammenfassung**

Eine unbekannte, reale, statische Raumsituation aus geometrischen Primitiven wird mit einem 2D Laserscanner, der an einem Roboterarm befestigt ist, vermessen. Die Messdaten werden segmentiert, analysiert und ausgewertet, sodass jede geometrische Primitive aus dem vermessenen Raum erkannt wird und modelliert werden kann. Ziel der Arbeit ist eine gut approximierende 3D Modellierung eines realen Weltausschnittes nachzubilden.



# Vorwort

Diese Diplomarbeit wurde für den Arbeitsbereich TAMS - Technische Aspekte Multimodaler Aspekte - unter der Leitung von Prof. Dr. Jianwei Zhang erarbeitet. Forschungsschwerpunkt des Arbeitsbereiches TAMS ist die Untersuchung und Entwicklung von intelligenten Servicerobotern und kognitiven Systemen.

Intelligente Serviceroboter sollen menschenähnliche Verhaltensmuster aufweisen, wie z.B. Objekte erkennen und nach ihnen greifen, sich in unbekanntem Umgebungen bewegen können, auf menschliche Stimmen reagieren können, etc. Systeme mit menschenähnlichen Verhaltensmustern erfordern einen Informationsaustausch zwischen Roboter und Umgebung, d.h., ein Wissenstand bestehend aus dem Wissen über die Eigenschaften des Roboters und Wissen über die Roboterumgebung. Das Wissen über die Robotereigenschaften liegt meist vor und ist erweiterbar. Wissen über die Roboterumgebung muss immer wieder aktualisiert werden, aufgrund von Lageveränderungen des Roboters und Umwelteinflüssen. Sensoren werden zur Informationsgewinnung aus Roboterumgebungen verwendet. Die Sensordaten werden durch Algorithmen interpretiert und ausgewertet. Die Interpretation und Auswertung werden genutzt, sodass der Roboter eine interaktionsfähige Handlung ausführen kann.

In dieser Arbeit geht es um die Entwicklung eines technischen Seh- bzw. Wahrnehmungssystems für den TAMS Service Roboter TASER.

---

## Hinweise

In dieser Arbeit werden einige Grundlagen, die in den meisten Standardwerken zu dem Thema 3D Darstellung und 3D Modellierung sich überschneiden, nicht referenziert. Diese Grundlagen können in den meisten Publikationen nachgeschlagen werden. Außerdem habe ich einige Begrifflichkeiten in dieser Arbeit bei deren englischer Nomenklatur beibehalten, da sie sich im deutschen Fachkollegium mit ihrer englischen Bezeichnung etabliert haben und meist auch nicht vernünftig und präzise übersetzbar sind.

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
1.1	Motivation und Ziel . . . . .	1
1.2	Verwandte Arbeiten . . . . .	4
<b>2</b>	<b>Theoretische Grundlagen und Stand der Technik</b>	<b>7</b>
2.1	Optische Messung - Lasertechnologie . . . . .	7
2.1.1	Laser . . . . .	7
2.1.2	Laserscanner und Messprinzipien . . . . .	8
2.1.3	Laserscanning . . . . .	11
2.1.4	2D Laserscanner Scanmethoden . . . . .	13
2.2	Roboterarm . . . . .	14
2.2.1	Einsatz von Roboterarmen . . . . .	14
2.2.2	Kinematik Roboterarm . . . . .	14
2.3	Grundlagen dreidimensionaler Darstellungen . . . . .	16
2.3.1	Koordinatensysteme . . . . .	16
2.3.2	Geometrische Transformationen im dreidimensionalen Raum	19
2.3.3	Analytische Geometrie im zweidimensionalen Raum . . . . .	21
2.4	Geometrische Modellierung . . . . .	23
2.4.1	Geometrische Primitive . . . . .	23
2.4.2	3D Grundkörper . . . . .	24
2.4.3	Zweidimensionale Quadriken . . . . .	26
2.5	Anwendungen auf Punktwolken . . . . .	27
2.5.1	Triangulation . . . . .	27
2.5.2	Clusteranalyse . . . . .	28
2.6	Verwendete Software . . . . .	30
2.6.1	Java Native Interface - JNI . . . . .	30
2.6.2	Java 3D . . . . .	30

<b>3 Hardware</b>	<b>33</b>
3.1 Roboterplattform . . . . .	33
3.2 PA10-6C Roboterarm und BarrettHand Dreifingerhand . . . . .	36
3.3 HOKUYO URG-LX04 . . . . .	38
<b>4 Entwurf und Implementierungsdetails des URG-PA10 3D Lasersystems</b>	<b>41</b>
4.1 Datenerfassung - Scanmethode (Modul 1) . . . . .	42
4.1.1 Entwurf der linearen Scanmethode . . . . .	46
4.2 Clusteranalyse zur Segmentierung der 3D Punktwolke (Modul 2) . . . . .	51
4.3 Wissensextraktion und Modellierung (Modul 3) . . . . .	60
4.3.1 Auswertung der Tischlinienwolke . . . . .	60
4.3.2 Analyse und Auswertung der markierten Objektlinienwolke und Verdeckungslinienwolke . . . . .	61
4.3.3 Modellierung unbekannter Objekte . . . . .	72
<b>5 Implementierung des URG-PA10 3D Lasersystems</b>	<b>75</b>
5.1 Implementierung der Scanmethoden . . . . .	77
5.2 Datenverarbeitung . . . . .	80
<b>6 Experimentelle Ergebnisse und Auswertung</b>	<b>83</b>
6.1 Experimente und Ergebnisse der Scan-Mit-Stop Methode . . . . .	85
6.2 Experimente und Ergebnisse der Scan-Ohne-Stop Methode . . . . .	87
6.3 Vergleich der beiden Scanmethoden . . . . .	89
<b>7 Fazit und Ausblick</b>	<b>93</b>
<b>Literaturverzeichnis</b>	<b>95</b>
<b>Erklärung</b>	<b>99</b>



# Kapitel 1

## Einleitung

### 1.1 Motivation und Ziel

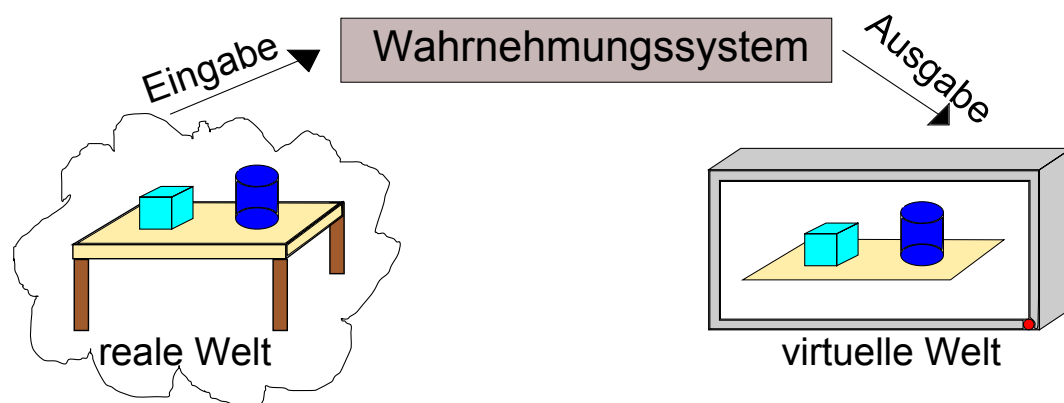


Abbildung 1.1: Skizze des hier zu entwickelnden Wahrnehmungssystems.

Im Rahmen dieser Arbeit wird ein laserbasiertes Wahrnehmungssystem für den TAMS Service Roboter TASER entwickelt.

Ziel des Wahrnehmungssystems ist dem TASER Informationen über eine reale Raumsituation, die aus geometrischen Primitiven besteht, zu liefern und ein 3D Modell der Raumszene an einem Bildschirm auszugeben. Die Funktion des Systems ähnelt der Funktion des menschlichen Sehsystems, ist aber bei weitem nicht so komplex.

Das menschliche Sehsystem ist ein komplexes Gebilde, das dem Menschen hilft,

sich in seiner Umgebung zurecht zu finden. Viele Menschen, die sehen können, sind in der Lage Farben zu unterscheiden, Objekten Eigenschaften, wie Größe, Form, Material zuzuordnen, Entfernungen zu und zwischen Objekten abzuschätzen, statische und dynamische Umgebungen wahrzunehmen, etc. Das Wahrnehmungssystem dieser Arbeit ist auf die Erkennung einer real statischen Tischfläche, auf die Quader-Objekte und Zylinder-Objekte platziert werden, spezialisiert. Aufgabe des Wahrnehmungssystems ist, die einzelnen Objekte zu detektieren, jedem Objekt geometrische Eigenschaften zuzuweisen und die genaue Lage jedes Objekts im Raum zu bestimmen.

Informationen über die Tischszene werden für das menschliche Sehsystem über das Auge geliefert, und für das hier zu entwickelnde Wahrnehmungssystem wird ein 2D Laserscanner, der an einen Roboterarm angebracht wird, genutzt. Sowohl das Auge als auch der 2D Laserscanner sind die Schnittstelle zwischen der realen Welt und ihrem zugehörigen System.

Heutzutage werden in vielen Forschungsarbeiten insbesondere, im Bereich der Robotik, 2D Laserscanner zur Erfassung von Räumen und Gebieten eingesetzt. Der Einsatz von Laserscannern ist eine junge Technologie, die einem starken Wachstum unterliegt und seit 20 Jahren Anwendung in Forschung und Industrie findet. Gründe für die Beliebtheit von Laserscannern liegen darin, dass sie binnen weniger Sekunden tausende von Messwerten ermitteln und Objekte in Entfernungen bis zu mehreren Kilometern erfassen. Laserscanner werden an bewegliche Vorrichtungen, wie z.B. an einer Dreh- und Schwenkvorrichtung, die als Pan-Tilt-Unit (PTU) bezeichnet wird, oder mobilen Plattform angebracht. In dieser Arbeit wird erstmals ein 2D Laserscanner an einen Roboterarm montiert, um Informationen aus der Außenwelt aufzunehmen. Ziel ist eine geeignete Kombination aus dem 2D Laserscanner und einer vordefinierten Armbewegung zu realisieren, um eine reale Tischszene zu vermessen. Die Vermessungsmethode erzeugt eine 3D Punktwolke. Diese 3D Punktwolke ist eine unsortierte Menge von 3D Punkten, die jeweils einen Oberflächenpunkt der Tischszene beschreiben.

Für die Strukturierung der 3D Punktwolke wird eine Clusteranalyse entwickelt, die eine automatisierte Segmentierung der 3D Punktwolke ausführt. Das Ergebnis der Segmentierung ist eine Menge von Clustern. Jedes Cluster enthält eine Menge von Punkten, die zu genau einer Oberfläche der Tischszene gehört.

Die Anzahl der Cluster sollte gleich der Anzahl der Objekte plus der Tischfläche sein. Allgemein wird eine Clusteranalyse auf eine heterogene Menge angewandt, um die Menge in homogene Gruppen zu unterteilen. Clusteranalysen werden in vielen Bereichen, insbesondere Wirtschaft, Raumordnung, Umwelt, Gesundheitswesen, Verkehr, etc. genutzt.

Die aus der Clusteranalyse entstehenden Cluster werden einzeln analysiert, ausgewertet und modelliert. Bei der Analyse und Auswertung wird jedes Cluster auf bestimmte geometrische Eigenschaften bzw. Parameter überprüft, außer dem Cluster, der die Tischoberflächenpunkte beinhaltet. Die Tischfläche ist als ein Quader vorgegeben. Anhand der Analyse- und Auswertungsergebnisse wird jedes einzelne Objekt modelliert und lokalisiert.

Das Endergebnis ist ein 3D Modell der Tischszene. Aus diesem 3D Modell können bestimmte Aussagen abgeleitet werden, wie z.B.

- genaue Entfernungsangaben zu und zwischen den Objekten oder zwischen Objekt und Roboter
- Berechnung der Volumen der Objekte und daraus schlussfolgernd sagen, welches Objekt am größten ist
- Bestimmung des höchsten Objekts,
- Identifizierung der Objekte, die den größten bzw. minimalsten Flächenanteil der Tischfläche verdecken
- geometrische Gemeinsamkeiten der Objekte erkennen
- etc.

Der klare Vorteil dieses Wahrnehmungssystems gegenüber dem menschlichen Wahrnehmungssystem ist, genauere numerische Angaben von Entfernungen und Größen der Objekte festzulegen. Das menschliche Sehsystem kann Entfernungen und Größen nur grob abschätzen.

Die Erzielung obiger Aussagen kann als Grundlage für ein Robotergreifsystem oder Überwachungssystem verwendet und erweitert werden.

## 1.2 Verwandte Arbeiten

Einzelne Module des hier vorgestellten Systems sind Gegenstand aktueller Forschungsarbeiten. Es liegen wenige wissenschaftliche Arbeiten vor, die Vermessungstechniken zur Erzeugung einer 3D Punktwolke beschreiben und darauf aufbauend eine Flächenrückführung auf die 3D Punktwolke durchführen. Im Folgenden werden Forschungsarbeiten vorgestellt, die dem aktuellen Stand der Technik entsprechen.

Ein Forschungsschwerpunkt in der heutigen Robotertechnologie ist, Räume mittels von 2D Laserscannern zu vermessen.

In der Arbeit von [1] werden vier rotationelle Methoden zur Erzeugung von 3D Punktwolken mittels eines 2D Laserscanners beschrieben. In jeder Methode wird der 2D Laserscanner in einer bestimmten Haltung um eine Achse gedreht. Das Ergebnis der Methoden sind 3D Punktwolken mit jeweils unterschiedlichen Punktedichteverteilungen.

Ein wichtiger Punkt bei der Erzeugung von 3D Punktwolken mittels eines 2D Laserscanners ist die Registrierung. Aufgabe der Registrierung ist das korrekte Überführen von Scanwerten aus unterschiedlichen Perspektiven in einen gemeinsamen Raum. Die meisten Algorithmen, die für die Registrierung genutzt werden, basieren auf dem Iterative-Closest-Point Algorithmus (ICP).

In [2] werden Registrierungsalgorithmen basierend auf dem ICP-Algorithmus vorgestellt, um mehrere 3D Scans, die auf einer mobilen Roboterplattform erzeugt werden, in ein gemeinsames dreidimensionales Koordinatensystem zu überführen und zusätzlich die Lage des Roboters im Raum zu bestimmen.

In [3] ist ein Registrierungsmodul beschrieben. Dieses Registrierungsmodul registriert die Position eines 2D Laserscanners, der sich auf einer mobilen Plattform befindet.

Ein weiterer zentraler Forschungsschwerpunkt in der heutigen Entwicklung von autonomen Robotern und in der Computergraphik ist die Flächenrückführung aus einer 3D Punktwolke. Dabei gibt es unterschiedliche Ansätze, wie man aus einer 3D Punktwolke eine Oberfläche rekonstruieren kann. Die meisten Techniken und Algorithmen für die Punktwolkenverarbeitung sind aus der Bildverarbeitung.

In der Arbeit von [4] werden sechs Linialgorithmen aus der Bildverarbeitung eingesetzt, um Linien aus 2D Punktwolken zu extrahieren. Für die Bearbeitung von 2D Punktwolken wird in der Arbeit von [5] die Methode der kleinsten Quadrate genutzt. Die Methode der kleinsten Quadrate ist ein mathematisches Verfahren zur Festlegung von Linien und Kreisen aus einer gegebenen Punktwolke und ist robust gegenüber kleinen Messfehlern bzw. Messabweichungen.

Die Arbeit von [6] beschäftigt sich mit der Extraktion von Oberflächen aus einer gegebenen 3D Punktwolke und unterscheidet zwischen zwei Techniken, Segmentierung und direkte Extraktion. Die Segmentierung zerlegt eine 3D Punktwolke aufgrund ähnlicher Merkmalseigenschaften in Gruppen. Die direkte Extraktion nimmt eine Reihe von Punkten aus der Punktwolke und überprüft, ob die Punkte bestimmte Parameterwerte erfüllen. Ein direktes Extraktionsverfahren eignet sich hauptsächlich für 3D Punktwolken von Szenen, die aus geometrischen Primitiven bestehen. Denn geometrische Primitive sind durch wenige Parameter beschreibbar.

Die Arbeit [7] von Alferes beschreibt die 3D Datenaquirierung einer Umgebung mit Hilfe eines 2D Laserscanners, der auf einer mobilen Plattform ist. Die Umgebung ist aus planaren Flächen aufgebaut. Daher ist die Verarbeitung der 3D Punktwolke speziell auf die Findung von Punktemengen konzentriert, die sich auf einer Ebene befinden.

Planare Flächen werden auch in der Arbeit von [8] zur Approximation einer 3D Punktwolke verwendet. Der Algorithmus segmentiert eine 3D Punktwolke in kleine Gruppen und trianguliert die Punkte, sodass ein Dreiecksnetz entsteht. Befinden sich innerhalb einer Gruppe zwei benachbarte Dreiecke, die in einer Ebene liegen, so werden diese Dreiecke zu einem Dreieck zusammengefasst. Der Algorithmus endet, sobald keine benachbarten Dreiecke einer Gruppe in einer Ebene liegen.



# Kapitel 2

## Theoretische Grundlagen und Stand der Technik

### 2.1 Optische Messung - Lasertechnologie

#### 2.1.1 Laser

Laser steht für light amplification by stimulated emission of radiation und beschreibt den Vorgang der Lichtverstärkung durch stimulierte Emission. Dieser Vorgang führt zur Entstehung der Laserstrahlung. Anfang des 20. Jahrhunderts hat Albert Einstein die Idee der stimulierten Emission eingeführt, und 1960 hat Theodore Maiman den ersten funktionsfähigen Laser realisiert. Im Gegensatz zu anderen Lichtquellen, wie Glühlampe oder Sonne, die Licht in alle Raumrichtungen mit relativ unbestimmten Frequenzen aussenden, emittiert der Laser einen gut gebündelten Strahl mit hoher Frequenzschärfe. Allgemein zeichnet sich Laserlicht durch folgende Eigenschaften aus [9]:

- geringe spektrale Linienbreite
- starke Bündelung
- hohe Strahlintensität(-energie)
- Eignung zur Erzeugung ultrakurzer Lichtimpulse

### 2.1.2 Laserscanner und Messprinzipien

Der Laserscanner ist ein optisches Messgerät, der Entfernungen zu reflektierenden Objekten ermittelt. Der Laserscanner sendet einen Laserstrahl aus. Der Laserstrahl trifft auf eine Objektoberfläche auf und wird von der Objektoberfläche zurückgeworfen. Um die Entfernung zu berechnen, gibt es unterschiedliche Messprinzipien:

#### Time-of-Flight

Der Laserscanner emittiert einen Laserstrahl, der von einem Gegenstand zum Scanner zurückreflektiert wird. Bei diesem Verfahren wird die gesamte Zeit gemessen, die der Laserstrahl vom Aussendezeitpunkt bis zum Empfangszeitpunkt benötigt, siehe Abbildung 2.1. Der Abstand wird aus der Zeit und der bekannten Geschwindigkeit des Laserstrahls bestimmt. Da die gemessene Zeit eines Laserstrahls nicht immer genau und richtig sein kann, werden meist mehrere Laserstrahls gesendet, um dann einen Mittelwert der Zeit zu ermitteln.



Abbildung 2.1: Time-of-Flight.

#### Triangulation

Die Triangulationsmethode kann als Lichtschnittverfahren bezeichnet werden. Der Laserscanner projiziert einen Laserstrahl auf ein Messobjekt. Abhängig von der Entfernung des Messobjektes wird der Laserstrahl unter einem bestimmten Winkel auf einen Detektor, z.B. CCD-Kamera, reflektiert. Die Entfernung wird mit der bekannten Grundlinie  $b$  und den zwei anliegenden Winkeln  $\alpha$  und  $\beta$ , siehe Abbildung 2.2, berechnet.



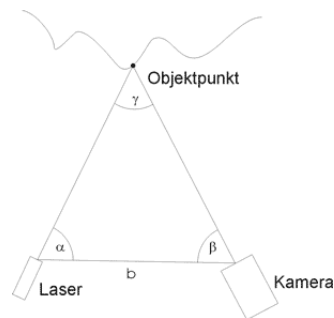


Abbildung 2.2: Triangulation aus [10].

## Phasenmessverfahren

Zur Distanzmessung wird der Phasenunterschied aus dem gesendeten und dem empfangenen Lichtsignal benutzt, siehe Abbildung 2.3.

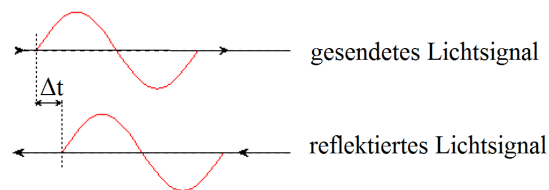


Abbildung 2.3: Phasenmessverfahren.

## 1D Laserscanner

Ein 1D Laserscanner ist ein punktuell Messsystem, das als Abstandsmesser bezeichnet wird. 1D Laserscanner sind mit einem Laserstrahl und einem Detektor ausgerüstet und werden in vielen Bereichen eingesetzt, wie z.B.

- zur Ermittlung von Bewegungen und Veränderungen von Geländen oder anderen Objekten
- Barcodeleser

Ein 1D Laserscanner ist in Abbildung 2.4 zu sehen.

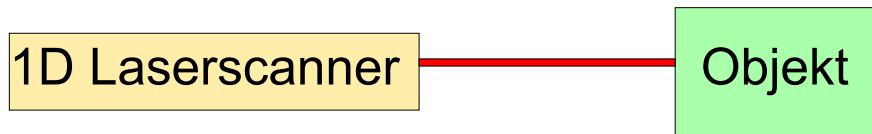


Abbildung 2.4: 1D Laserscanner.

## 2D Laserscanner

Ein 2D Laserscanner ist ein 1D Laserscanner, der genau einen Laserstrahl emittiert. Mittels eines rotierenden Umlenkspiegels wird der Laserstrahl über einen bestimmten Öffnungswinkel gelenkt. Kennzeichnende Eigenschaften von 2D Laserscannern sind:

- Messweite
- Öffnungswinkel
- Winkelschrittweite

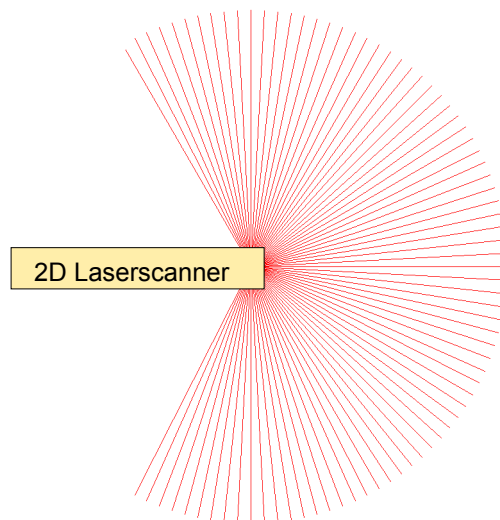


Abbildung 2.5: 2D Laserscanner.

## 3D Laserscanner

Ein 3D Laserscanner wirft ebenfalls einen Laserstrahl aus, der durch eine bestimmte Vorrichtung horizontal und vertikal in festgesetzten Winkelschritten bewegt wird, um 3D Daten einer Objektoberfläche zu erfassen.

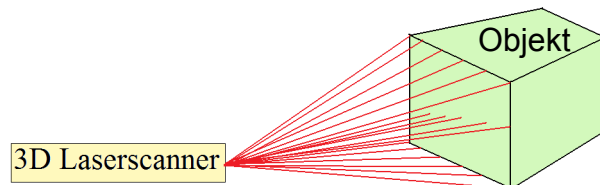


Abbildung 2.6: 3D Laserscanner.

### 2.1.3 Laserscanning

Der Begriff Laserscanning ist ein Akronym für Laserabtastung im dreidimensionalen Bereich und spezifiziert die Art und Weise, wie Objektoberflächen mittels eines Laserscanners abgetastet werden. In [11] und anderen Arbeiten unterscheidet man zwischen folgenden drei Formen des Laserscannings:

- Airborne Laserscanning
- Terrestrial Laserscanning
- Mobile Laserscanning

### Airborne Laserscanning

Beim Airborne Laserscanning Verfahren ist das Laserscansystem auf einem Flugzeugträger, z.B. Helikopter, Gleitschirm-Ultraleichtfluggerät, integriert. Ziel von Airborne Laserscansystemen ist die Aufnahme von Landschaftsoberflächen, um Höhen und Strukturen von Geländen bzw. Erdoberflächen zu erkennen, siehe Abbildung 2.7.

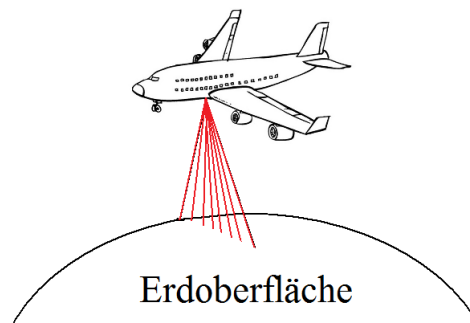


Abbildung 2.7: Airborne Laserscanning.

## Terrestrial Laserscanning

Das Terrestrial Laserscanning ist ein bodengestütztes Laserverfahren, siehe Abbildung 2.8. Der Laserscanner erfasst einen bestimmten Bereich einer Umgebung von einem fixen Standpunkt aus.

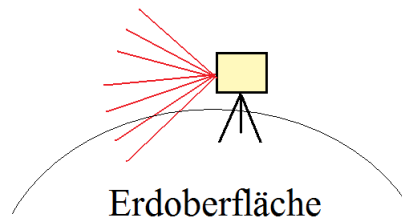


Abbildung 2.8: Terrestrial Laserscanning.

## Mobile Laserscanning

Beim Mobile Laserscanning befindet sich der Laserscanner auf einer mobilen Plattform, um 3D Daten aus einem bestimmten Raumbereich zu erfassen. Der Standort des Laserscanners wird durch die Bewegung der mobilen Plattform verändert, siehe Abbildung 2.9.

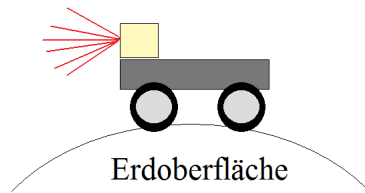


Abbildung 2.9: Mobile Laserscanning.

### 2.1.4 2D Laserscanner Scanmethoden

In der Arbeit von [1] werden vier verschiedene Scanmethoden vorgestellt, die in Abbildung 2.10 zu sehen sind. Die Scanmethoden erzeugen 3D Punktwolken mit einem 2D Laserscanner, der einen Öffnungswinkel (Detection Angle) von  $180^\circ$  hat. In allen vier Scanmethoden wird der Laserscanner um eine Achse gedreht und alle Methoden haben gemeinsam, dass die Punktedichte Verteilung immer maximal an der Rotationsachse und minimal senkrecht zur Rotationsachse ist. Jede Scanmethode eignet sich für unterschiedliche Anwendungen, z.B. Pitching Scan und Rolling Scan eignen sich bei einer Drehung von  $360^\circ$  zur Vermessung von Räumen, das Rolling Scan und Yawing Top Scan eignen sich zur Erfassung von Einzelobjekten. Diese vier Methoden können als rotationelle Scanmethoden klassifiziert werden.

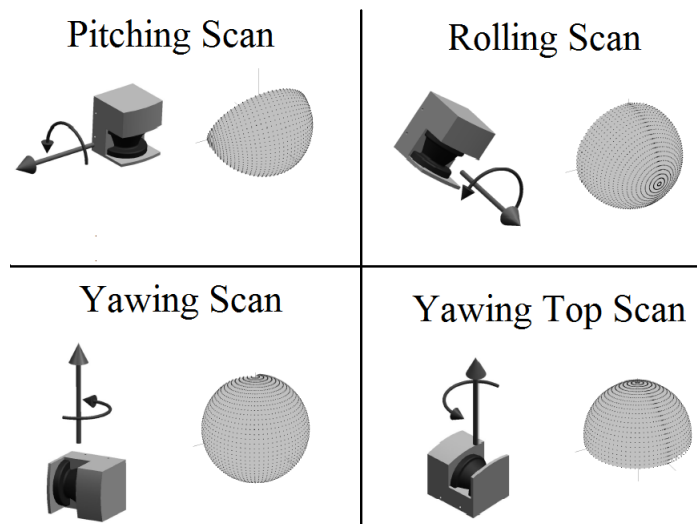


Abbildung 2.10: Rotationelle Scanmethoden.

## 2.2 Roboterarm

### 2.2.1 Einsatz von Roboterarmen

Roboterarme werden heutzutage in vielen Bereichen, wie z.B. in der Industrie und insbesondere in der Raumfahrttechnik, verwendet. Besonders in kritischen Bereichen oder aus Sicherheitsgründen werden Roboterarme gern eingesetzt, wie z.B. zur Reparatur beschädigter Rohre in einem Kernkraftwerk. In der Raumfahrt dienen Roboterarme zur Durchführung von Reparatur- und Wartungsarbeiten an Satelliten. Für die Exploration von anderen Planeten, wie dem Mars, wird der Roboterarm zur Entnehmung von Bodenproben genutzt.

Die Roboterarme sind in den meisten Bereichen ferngesteuert oder sie durchlaufen einen festvordefinierten Bewegungsablauf ab.

### 2.2.2 Kinematik Roboterarm

Die Kinematik beschreibt den mechanischen Aufbau des Roboterarmes. Für jeden Anwendungsbereich sind die kinematischen Eigenschaften eines Roboterarmes unterschiedlich. Der Roboterarm ist eine Kombination aus Gelenken und Komponenten.

Im Skript [12] werden vier Merkmale zu den Komponenten  $L_i$  und Gelenken  $J_i$  eines Roboterarmes beschrieben:

1. Komponentenlänge  $a_i$ : Betrachte die zwei gegenüberliegenden Gelenke der Komponente  $L_i$ . Es gibt eine Gerade, die auf beiden Gelenkachsen  $J_i$  und  $J_{i+1}$  senkrecht steht. Nimm die Länge  $a_i$  dieser Gerade.
2. Komponentenverdrehung  $\alpha_i$ : Betrachte die zu beiden Gelenkachsen senkrecht stehende Gerade als Normale einer Ebene. Projiziere beide Achsen auf die Ebene und nimm Winkel  $\alpha_i$  zwischen den beiden.
3. Komponentenverschiebung  $d_i$ : Zwei benachbarte Komponenten  $L_{i-1}$  und  $L_i$  sind gegeneinander verschoben, haben aber eine gemeinsame Achse  $J_i$ . Aufgrund der Komponentenlängen von  $L_{i-1}$  und  $L_i$  werden auf der Achse  $J_i$  zwei Punkte  $P_1$  und  $P_2$  definiert. Nimm den Abstand  $d_i$  zwischen beiden als Verschiebung der Komponenten.

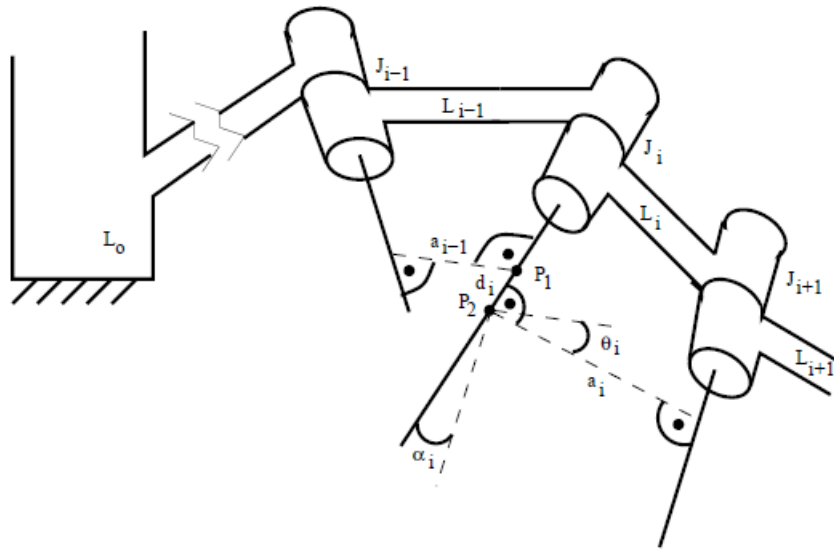


Abbildung 2.11: Aufbau eines Roboterarmes aus [12].

4. Komponentenrotation  $\Theta_i$ : Verlängere die Gerade (mit Komponentenlänge  $a_{i-1}$ ) über Gelenkachse  $J_i$  hinaus und verschiebe sie nach Punkt  $P_2$ . Nimm den Winkel  $\Theta_i$  zur Geraden mit Komponentenlänge  $a_i$  als Komponentenrotation.

Eine gute Einführung in die Thematik von Mechaniken und Steuerungen von Robotern ist in [13] gegeben.

## 2.3 Grundlagen dreidimensionaler Darstellungen

### 2.3.1 Koordinatensysteme

#### Zweidimensionales Polarkoordinatensystem

Ein zweidimensionales Polarkoordinatensystem besteht aus zwei Geraden, die orthogonal zueinander liegen (Abbildung 2.12). Ein Punkt in einem Polarkoordinatensystem wird durch ein zweier Tupel  $(R, \theta)$  beschrieben, siehe Abbildung 2.13.

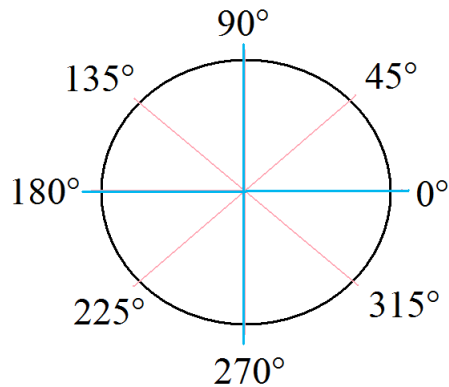


Abbildung 2.12: Polarkoordinatensystem.

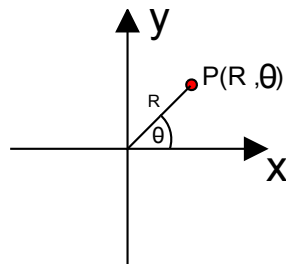


Abbildung 2.13: Punkt in einem Polarkoordinatensystem.



## Zweidimensionales kartesisches Koordinatensystem

Ein zweidimensionales kartesisches Koordinatensystem besteht aus zwei Achsen, die wie beim Polarkoordinatensystem orthogonal zueinander liegen. Ein Punkt in einem zweidimensionalen Koordinatensystem wird durch ein zweier Tupel  $(x, y)$  beschrieben, siehe Abbildung 2.14.

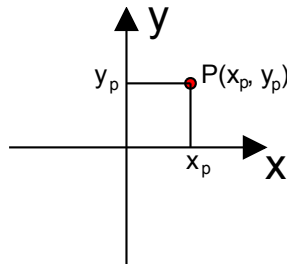


Abbildung 2.14: Punkt in einem zweidimensionalen kartesischen Koordinatensystem.

## Dreidimensionales kartesisches Koordinatensystem

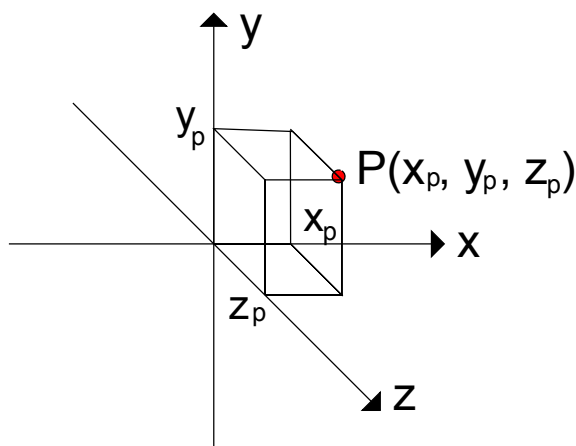


Abbildung 2.15: Punkt in einem dreidimensionalen Koordinatensystem.

Ein dreidimensionales Koordinatensystem besteht aus drei Achsen, die alle zueinander orthogonal liegen. Ein Punkt in einem dreidimensionalen Koordinatensystem wird durch ein dreier Tupel  $(x, y, z)$  beschrieben, siehe Abbildung 2.15.

### 2.3.1.1 Umrechnung von Polarkoordinaten in kartesische Koordinaten

Um einen Punkt  $(R, \theta)$  eines zweidimensionalen Polarkoordinatensystems in einen kartesischen Punkt  $(x, y)$  umzurechnen, werden folgende Formeln für die Berechnung des x und y Wertes genutzt:

$$x = R * \cos(\theta) \quad (2.1)$$

$$y = R * \sin(\theta) \quad (2.2)$$

### Zylinderkoordinatensystem

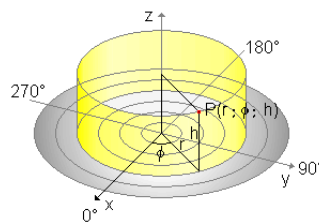


Abbildung 2.16: Zylinderkoordinatensystem aus [14].

Das Zylinderkoordinatensystem wird in dieser Arbeit zur Überführung von zweidimensionalen Polarkoordinaten in dreidimensionale kartesische Koordinaten benutzt. Zylinderkoordinaten sind ebene Polarkoordinaten, die um eine dritte Koordinate ergänzt sind. Diese dritte Koordinate beschreibt die Höhe eines Punktes senkrecht über (oder unter) der Ebene des Polarkoordinatensystems und wird im Allgemeinen mit  $z$  bezeichnet, siehe Abbildung 2.16.

### 2.3.2 Geometrische Transformationen im dreidimensionalen Raum

Geometrische Transformationen werden zur Positionierung, Verschiebung, Drehung oder Verformung von Objekten verwendet.

Drei wichtige geometrische Transformationen im zwei- und dreidimensionalen Raum sind

- Translation,
- Skalierung,
- Rotation.

Im folgenden sind die mathematischen Beschreibungen aus [15] entnommen.

Ein Punkt des  $\mathbb{R}^3$  wird durch vier Koordinaten  $(\tilde{x}, \tilde{y}, \tilde{z}, \omega)$  mit  $\omega \neq 0$  repräsentiert. Dabei beschreibt das vierer Tupel  $(\tilde{x}, \tilde{y}, \tilde{z}, \omega)$  den Punkt  $(\frac{\tilde{x}}{\omega}, \frac{\tilde{y}}{\omega}, \frac{\tilde{z}}{\omega}) \in \mathbb{R}^3$ . Der Punkt  $(x, y, z) \in \mathbb{R}^3$  kann in homogenen Koordinaten in der Form  $(x, y, z, 1)$  dargestellt werden.

In homogenen Koordinaten lässt sich eine Translation um den Vektor  $(d_x, d_y, d_z)$  als Matrixmultiplikation in der Form

$$\begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & d_x \\ 0 & 1 & 0 & d_y \\ 0 & 0 & 1 & d_z \\ 0 & 0 & 0 & 1 \end{pmatrix} * \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = \begin{pmatrix} x + d_x \\ y + d_y \\ z + d_z \\ 1 \end{pmatrix} \quad (2.3)$$

mit der Translationsmatrix:

$$T(d_x, d_y, d_z) = \begin{pmatrix} 1 & 0 & 0 & d_x \\ 0 & 1 & 0 & d_y \\ 0 & 0 & 1 & d_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (2.4)$$

darstellen.

Eine Skalierung um die Faktoren  $s_x, s_y, s_z$  ist durch

$$\begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} = \begin{pmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} * \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = \begin{pmatrix} s_x * x \\ s_y * y \\ s_z * z \\ 1 \end{pmatrix} \quad (2.5)$$

mit der Skalierungsmatrix:

$$S(s_x, s_y, s_z) = \begin{pmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (2.6)$$

gegeben.

Im dreidimensionalen Raum muss eine Rotationsachse, um den das Objekt rotiert werden soll, vorgegeben werden. Im zweidimensionalen Raum reicht ein Punkt für die Durchführung einer Rotation aus. Die drei elementaren Rotationen im 3D Raum sind die Rotationen um die Koordinatenachsen. Eine Rotation um die z-Achse um den Winkel  $\theta$  läßt sich folgendermaßen in homogenen Koordinaten beschreiben:

$$\begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} = \begin{pmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & -\cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} * \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} \quad (2.7)$$

Mit der Rotationsmatrix:

$$R_z(\theta) = \begin{pmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & -\cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (2.8)$$

Bei einer Rotation um die z-Achse ändert sich die z-Koordinate nicht. Die Matrizen für Rotationen um die x- und y-Achse erhält man aus der obigen Matrix, indem man die Rollen der Achsen entsprechend vertauscht, sodass eine Rotation um die x-Achse um den Winkel  $\theta$  durch die Matrix

$$R_x(\theta) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta & 0 \\ 0 & \sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (2.9)$$

und eine Rotation um die y-Achse um den Winkel  $\theta$  durch die Matrix

$$R_y(\theta) = \begin{pmatrix} \cos\theta & 0 & \sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (2.10)$$

realisiert wird.

Transformationen können als Wechsel des Koordinatensystems aufgefasst werden.

### 2.3.3 Analytische Geometrie im zweidimensionalen Raum

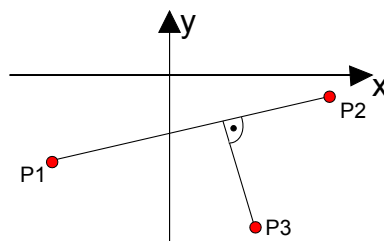


Abbildung 2.17: Kürzester Abstand zwischen dem Punkt P3 und der Linie  $P1\bar{P}2$ .

Einige der folgenden Gleichungen sind aus dem Buch [16] entnommen worden.

1. Der Abstand  $\mathbf{d}$  zwischen zwei Punkten  $P_1(x_1, y_1)$  und  $P_2(x_2, y_2)$  wird mit folgender Gleichung, die als euklidischer Abstand bezeichnet wird, berechnet:

$$\mathbf{d} = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \quad (2.11)$$

2. Der Mittelpunkt  $P_m$ , der zwischen den zwei Punkten  $P_1$  und  $P_2$  liegt, wird mit folgender Gleichung berechnet:

$$P_m = \left( \frac{x_1 + x_2}{2}, \frac{y_1 + y_2}{2} \right) \quad (2.12)$$

3. Der kürzeste Abstand zwischen einem Punkt und einer Gerade ist gegeben, wenn ein senkrecht Lot vom Punkt zur Geraden fällt, wie in Abbildung 2.17. Die Gleichung der Linie  $P_1\bar{P}_2$  ist wie folgt definiert:

$$P = (P_1 + u(P_2 - P_1)) \quad (2.13)$$

Die eingezeichnete Gerade von  $P_3$  zur Linie  $P_1\bar{P}_2$  und die Linie  $P_1\bar{P}_2$  stehen orthogonal zueinander, daher ist das Skalarprodukt beider Vektoren gleich null:

$$(P_3 - P)(P_2 - P_1) = 0 \quad (2.14)$$

Die Substitution von  $P$  durch die Formel (2.13) ergibt folgende Gleichung:

$$[P_3 - P_1 - u(P_2 - P_1)](P_2 - P_1) = 0 \quad (2.15)$$

Wird die Gleichung nach  $u$  aufgelöst, so ergibt sich:

$$u = \frac{(x_3 - x_1)(x_2 - x_1) + (y_3 - y_1)(y_2 - y_1)}{\|p_2 - p_1\|^2} \quad (2.16)$$

$u$  wird in die Liniengleichung eingesetzt und die  $x$  und  $y$  Werte des Punktes, der auf der Linie liegt und mit  $P_3$  eine Linie senkrecht zur Geraden bildet, berechnet:

$$x = x_1 + u(x_2 - x_1) \quad (2.17)$$

$$y = y_1 + u(y_2 - y_1) \quad (2.18)$$

Der Abstand von  $P_3$  zur Linie ergibt sich aus dem Abstand zwischen  $P_3$  und den errechneten  $x, y$ -Werten.

## 2.4 Geometrische Modellierung

Die geometrische Modellierung ist ein Teilbereich des Computer Aided Design (CAD), der sich mit der Theorie, den Techniken und den Systemen für die rechnergestützte Beschreibung und Darstellung von Raumkurven, Raumflächen und Körpern befasst [17]. Die geometrische Modellierung schafft Grundlagen u.a. für:

- die Berechnung geometrischer Eigenschaften von Körpern (Volumen, Oberfläche, usw.)
- graphische Darstellung von Körpern
- die Erzeugung realistischer Bilder, usw.

In dieser Arbeit erfolgt die geometrische Modellierung der realen Tischszene aus geometrischen Primitiven, da die Tischszene selbst nur aus primitiven Grundkörpern aufgebaut ist.

### 2.4.1 Geometrische Primitive

Geometrische Primitive sind Punkte, Linien und Flächen.

In [18] werden geometrische Primitive als einfache 2D und 3D Grundkörper bezeichnet, deren Größe und Form über Parameter gesteuert werden können.

Ein 2D Grundkörper ist beispielsweise eine Linie, ein Kreis, ein Rechteck, ein Vieleck, etc. 2D Grundkörper werden als Fläche erster Ordnung bezeichnet. Eine Fläche erster Ordnung ist ein Polygonzug aus Linien, deren Punkte in einer bestimmten Reihenfolge festgelegt sind.

Beispiele für 3D Grundkörper sind Kugel, Kegel, Quader, Zylinder und Torus. Diese 3D Grundkörper werden als Quadrik oder Fläche zweiter Ordnung bezeichnet.

## 2.4.2 3D Grundkörper

Für diese Arbeit werden die Grundkörper Quader und Zylinder verwendet.

### Quader

Ein Körper ist genau dann ein Quader, wenn alle Kanten, wie in Abbildung 2.18, senkrecht zueinander stehen. Ein Quader hat 8 Ecken, 12 Kanten und 6 Flächen. Von den insgesamt 12 Kanten sind vier Kanten immer gleich lang und parallel. Die 6 Flächen sind Rechtecke.

Geometrische Parameter und Eigenschaften eines Quaders sind

- Höhe ( $h$ ),
- Breite ( $b$ ),
- Länge ( $l$ ),
- Grundfläche ( $G$ ),
- Volumen ( $V$ ) und
- Oberfläche ( $O$ ).

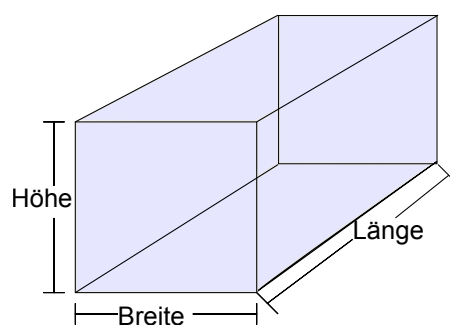


Abbildung 2.18: Quader.

In der Tabelle 2.1 stehen Formeln zur Berechnung bestimmter geometrischer Eigenschaften eines Quaders.



Geometrische Größe	Formel
Grundfläche (G)	$G = b * l$
Volumen (V)	$V = b * l * h$
Oberfläche (O)	$O = 2 * (b * l + b * h + l * h)$

Tabelle 2.1: Formeln zur Berechnung von geometrischen Größen eines Quaders.

## Zylinder

Ein Zylinder (siehe Abbildung 2.19) besteht aus zwei Kanten und drei Flächen (Grund- und Deckfläche, Mantelfläche). Die Grund- und Deckfläche sind ebene kreisförmige Flächen, die parallel zueinander liegen. Die Mantelfläche wird von parallelen Geraden gebildet. Liegen die Geraden senkrecht zur Grund- und Deckfläche, so wird der Zylinder als Kreiszyylinder bezeichnet.

Geometrische Parameter und Eigenschaften eines Kreiszyinders sind

- Höhe (h),
- Radius (r) der kreisförmigen Grund- und Deckfläche,
- Grundfläche (G),
- Mantelfläche (M)
- Volumen (V) und
- Oberfläche (O).

Geometrische Größe	Formel
Grundfläche (G)	$G = \pi * r^2$
Mantelfläche (M)	$M = (2 * \pi * r) * h$
Volumen (V)	$V = G * h$
Oberfläche (O)	$O = 2 * G + M$

Tabelle 2.2: Formeln zur Berechnung von geometrischen Größen eines Zylinders.

In der Tabelle 2.2 stehen Formeln zur Berechnung bestimmter geometrischer Eigenschaften eines Zylinders.

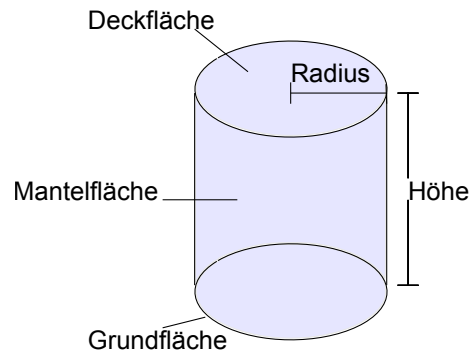


Abbildung 2.19: Zylinder.

## Komplexer Körper

Ein komplexer Körper wird von geometrischen Primitiven (Grundkörper), die mit booleschen Operatoren verknüpft werden, erzeugt. In Biedermans Arbeit [19] werden komplexe Körper bzw. komplexe Objekte von Menschen erkannt, indem diese komplexe Objekte in primitive Körper, wie z.B. Quader, Kugel, Zylinder, etc. zerlegt werden.

### 2.4.3 Zweidimensionale Quadriken

Quadrik im $\mathbb{R}^2$	Normalform
paralleles Geradenpaar	$-\frac{x^2}{r_x^2} + 1 = 0$
schneidendes Geradenpaar	$\frac{x^2}{r_x^2} - \frac{y^2}{r_y^2} = 0$
Kreis	$r_x = r_y, -\frac{x^2}{r_x^2} - \frac{y^2}{r_x^2} + 1 = 0$
Ellipse	$-\frac{x^2}{r_x^2} - \frac{y^2}{r_y^2} + 1 = 0$

Tabelle 2.3: Quadriken im  $\mathbb{R}^2$  mit ihrer Normalform.

Es existieren verschiedene Typen ebener Quadriken, in Abbildung 2.20 sind einige ebene Quadriken, die für diese Arbeit relevant sind, dargestellt. In Tabelle 2.3 sind einige ebene Quadriken mit ihrer Normalform aufgelistet. Ein Quadrik im  $\mathbb{R}^2$  ist von der Form

$$ax^2 + by^2 + cxy + dx + ey + f = 0 \quad (2.19)$$

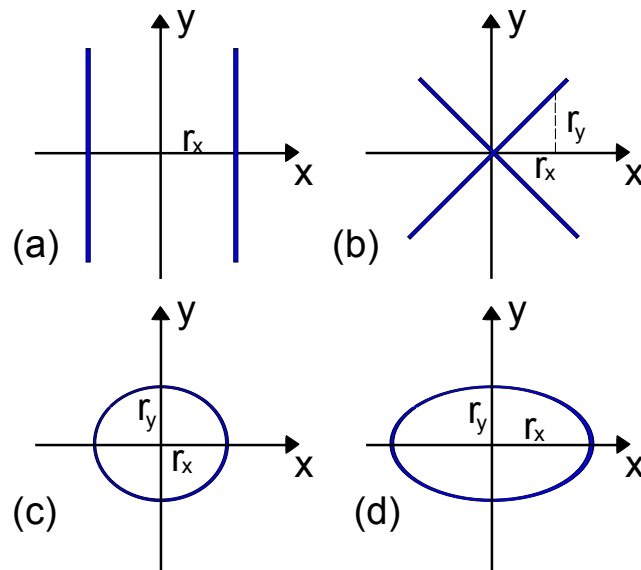


Abbildung 2.20: Quadriken im  $\mathbb{R}^2$  (a) Paralleles Geradenpaar. (b) Schneidendes Geradenpaar. (c) Kreis. (d) Ellipse.

## 2.5 Anwendungen auf Punktwolken

### 2.5.1 Triangulation

Unter dem Begriff Triangulation bezeichnet man in der Computergrafik eine Technik, die zur Flächenrückführung auf eine 3D Punktwolke angewandt wird. Die Dissertation von [20] beschäftigt sich mit unterschiedlichen Methoden von Verarbeitungstechniken, die auf 3D Punktwolke angewandt werden und beschreibt u.a. die Delaunay Triangulation. Die Delaunay Triangulation ist 1934 von dem russischen Mathematiker Boris Nikolajewitsch Delone erstmals eingeführt worden ist. Seither wird die Delaunay Triangulation in vielen Werken erwähnt und als Grundlage für Triangulationsverfahren eingesetzt. Die Delaunay Triangulation erfasst jeweils vier Punkte einer Punktwolke zu einem Tetraeder zusammen. Die Bestimmung der Tetraeder erfolgt nach dem folgenden Kriterium:

*Innerhalb der Kugel, die durch die vier Eckpunkte eines nicht entarteten Tetraeders eindeutig definiert ist, darf kein anderer Punkt in der Punktwolke liegen.*

Daraus erschließt sich, dass die Delaunay Triangulation eine Menge von Tetraedern ist und gleichzeitig die konvexe Hülle der gegebenen 3D Punktwolke definiert. Daher eignet sich dieses Verfahren nur für die Rekonstruktion der Oberfläche eines konvexen 3D Körpers.

*Ein räumlicher Körper heißt konvex, wenn je zwei beliebigen Punkten, die auf der Oberfläche des Körpers liegen, auch die Verbindungslinie dieser zwei Punkte vollständig innerhalb oder auf dem Rand des Körpers liegt.*

Beispiele für konvexe Körper sind z.B. Kugeln, Zylinder, Quader oder Pyramiden.

Weiterführende Literatur zum Thema Triangulation sind in dem Buch [21] und in der Dissertation [22].

### 2.5.2 Clusteranalyse

In [23] wird die Clusteranalyse als ein Gruppierungsverfahren bezeichnet, das eine große Objektmenge in Gruppen oder Klassen zerlegt. Diese Gruppen werden im englischsprachigen Raum als *Cluster* bezeichnet. Ziel der Clusteranalyse ist eine unsortierte Menge von Objekten zu analysieren. Die Objekte werden so in Clustern gruppiert, dass innerhalb eines Clusters die Objekte möglichst ähnlich und zwischen Clustern möglichst unähnlich sind [24]. Für die Ähnlichkeit zwischen den einzelnen Objekten hat das numerische Maß eine zentrale Bedeutung. Das Wort Ähnlichkeit kann durch die Wörter Gleichartigkeit und Benachbarkeit der Objekte ersetzt werden.

Anwendungsbereiche, in denen Clusteranalysen genutzt werden, sind in der

- Mustererkennung, z.B. Erdbeobachtungs- und Satellitenauswertung für ähnliche Landnutzung,
- Datenanalyse von wirtschaftlichen Aspekten,
- Bildverarbeitung.

In den meisten Büchern, die sich mit Clusterverfahren beschäftigen, werden eine Vielzahl an Clusteralgorithmen vorgeschlagen. Je nach Anwendungsgebiet und

Problemstellung werden in der Praxis Clusterverfahren speziell auf das zu lösende Problem erstellt. In der Literatur werden zwei Clusterverfahren grundsätzlich unterschieden:

- hierarchische Verfahren
- partitionierende Verfahren

Das hierarchische Verfahren erzeugt eine hierarchische Repräsentation der Datenmenge, während das partitionierende Verfahren die Datenmenge in eine bestimmte Anzahl von Clustern zerlegt [25]. Für diese Arbeit wird ein partitionierendes Verfahren eingesetzt, um die 3D Punktwolke zu segmentieren.

Folgende Definition und Gleichungen sind aus [23].

### Definition und formale Bezeichnungsweise

Mit  $E$  sei eine Objektmenge  $\{e_1, \dots, e_n\}$  und mit  $\mathfrak{R}(E)$  die Menge aller Teilmengen  $C \subseteq E$ , also die Potenzmenge von  $E$ , bezeichnet. Als einen Cluster kann man im Prinzip jede Teilmenge  $C \in \mathfrak{R}(E)$  bezeichnen.

Eine indizierte Menge von Clustern

$$L = \{C_1, \dots, C_m\} \subseteq \mathfrak{R}(E) \quad (2.20)$$

wird als Clusterung oder Gruppierung bezeichnet. Eine Clusterung ist ein Prozess der Gruppenbildung und ist disjunkt, wenn für alle  $i \neq j$

$$C_j \cap C_i = \emptyset \quad (2.21)$$

gilt, und sie heißt exhaustiv, wenn

$$\bigcup_{j \in \{1, \dots, m\}} C_j = E \quad (2.22)$$

ist.

Eine nichtdisjunkte Clusterung wird als überlappend, eine disjunkte und gleichzeitig exhaustive Clusterung als eine Partition oder Zerlegung von E bezeichnet.

Mehr zu Clusteranalysen sind in folgenden Büchern [26] und [27] zu finden.

Weiterführende Literaturen zur Bearbeitung von 2D und 3D Punktwolken sind in [28], [29] und [30].

## 2.6 Verwendete Software

### 2.6.1 Java Native Interface - JNI

Das Java Native Interface (JNI) ist eine standardisierte Anwendungsprogrammierschnittstelle (API), um aus der Programmiersprache Java Anwendungen bzw. Methoden, die in anderen Sprachen, wie z.B. C, C++ oder Assembler, geschrieben sind, zu verwenden. Mit Hilfe von JNI ist es Java möglich, Funktionen einer Windows-DLL oder einer shared library unter Linux, die in C oder C++ programmiert sind, aufzurufen.

### 2.6.2 Java 3D

Die Java 3D-API ist eine Klassenbibliothek für Java, um dreidimensionale Grafiken darzustellen. Dreidimensionale Körper, wie Quader, Kugel, Zylinder, etc. sind für den Anwender einfach zu erstellen, zu manipulieren und darzustellen. In Java 3D können virtuelle Welten realisiert werden, in denen man sich frei bewegen und mit seiner Umgebung interagieren kann. Java 3D benutzt und vereinigt die Konzepte existierender Low-Level-Grafiksysteme, wie z.B. OpenGL und Direct3D, und von Grafiksystemen auf höherer Abstraktionsebene, wie VRML oder Open Inventor. Das zentrale Konstrukt für die Anwendungsentwicklung in Java 3D ist der Szenengraph (Abbildung 2.21), der ebenfalls in Open Inventor und VRML als Datenstruktur verwendet wird. Der Szenengraph ist eine objektorientierte Datenstruktur, mit der die dreidimensionale Szene beschrieben wird. Objekte, Behavior (Verhalten von

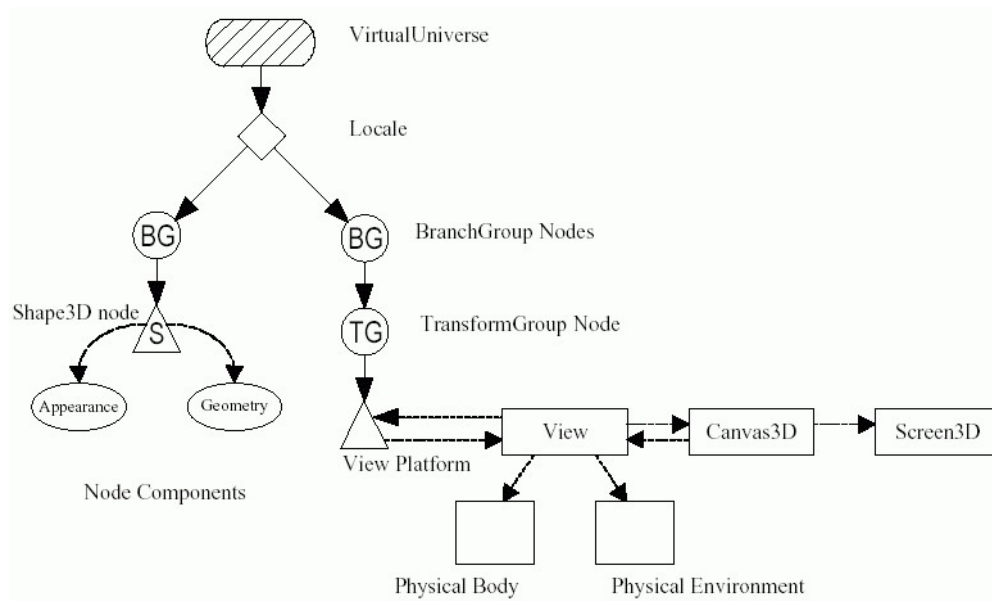


Abbildung 2.21: Java 3D Szenengraph [31].

Objekten), Licht und Tonquellen unterliegen im Szenengraph einer hierarchischen Ordnung. In Java 3D bildet die Orientierung der Raumachsen ein rechtshändiges Koordinatensystem.





# Kapitel 3

## Hardware

Die meisten Abbildungen in diesem Kapitel und das Wissen über den TASER und seinen Komponenten sind aus der Dissertation von [32] entnommen worden.

### 3.1 Roboterplattform

In diesem Projekt wird mit dem TAMS-Serviceroboter TASER gearbeitet, siehe Abbildung 3.1. Der Serviceroboter ist eine modifizierte Version des MP-L655 der Firma NEOBOTIX.



Abbildung 3.1: TAMS Serviceroboter - TASER.

Bestandteile des Roboters sind

- eine mobile Plattform mit einem Differenzialantrieb und Rad-Endcodern,
- ein Roboterarm PA10-6C der Firma Mitsubishi Heavy Industries und
- eine Dreifingerhand von Barrett Technologies,
- eine analoge Handkamera,
- ein Stereo-Kamerakopf bestehend aus einer Pan-Tilt-Unit (PTU) und zwei Sony DFW-VL5006 Firewire-Digitalkameras,
- ein Omnivisionsystem mit Sony DFW-SX9007 Firewire-Digitalkamera und hyperbolidem Spiegel und
- zwei SICK Lasermesssysteme.

Zur Steuerung der Motoren und Erfassung der Daten ist ein Pentium IV PC in die Plattform integriert. Der aktuelle Stand der Hardwarekomponenten des TASERs sind in Abbildung 3.2 zu sehen.

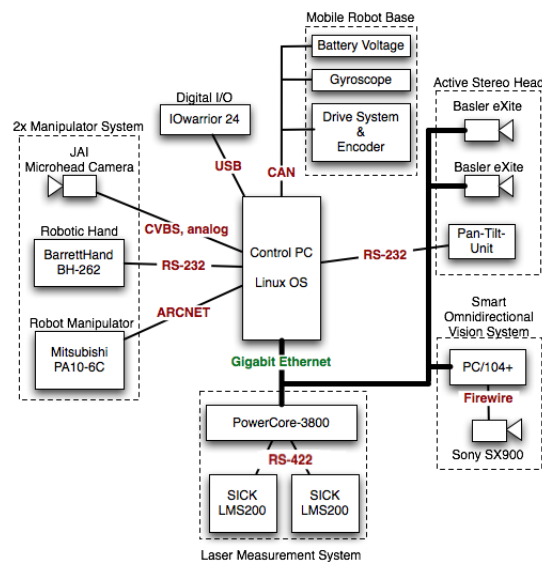


Abbildung 3.2: Aktueller Stand der Hardware des TASERs aus [33].

In Abbildung 3.3 ist die mobile Plattform mit dem Roboterarm und Bemaßung zu sehen.

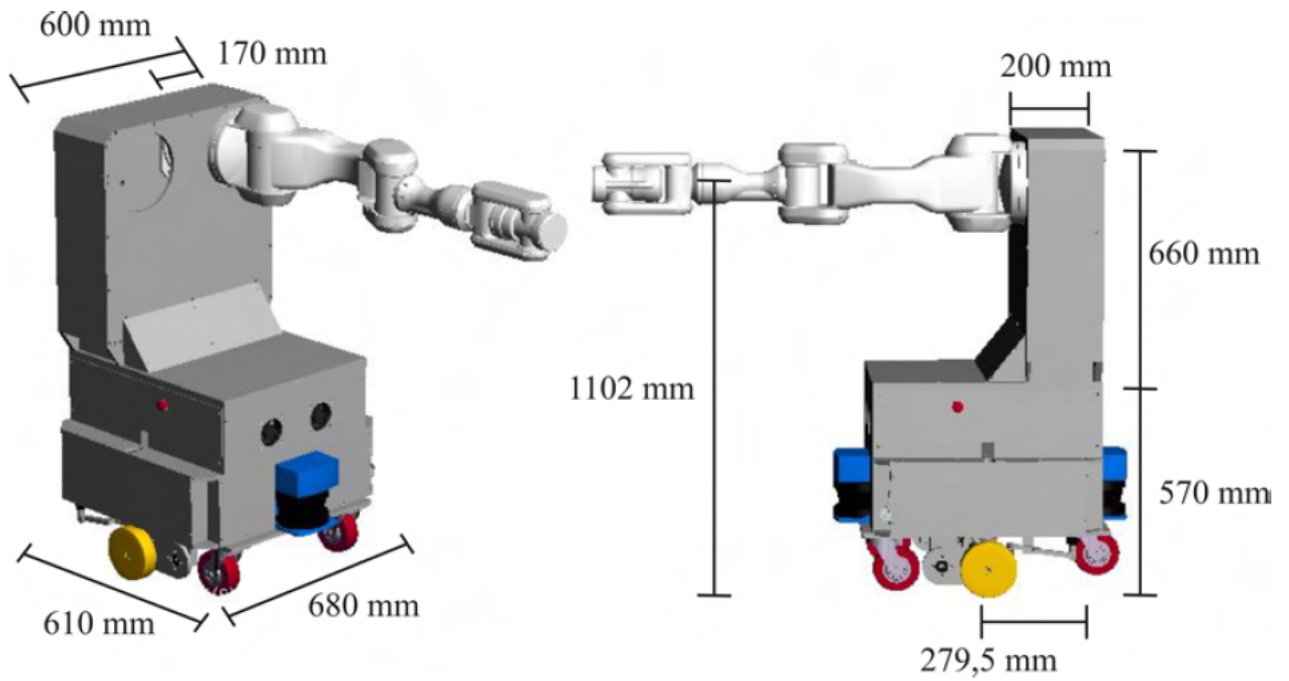


Abbildung 3.3: TAMS Serviceroboter mit Roboterarm und Bemaßung.

## 3.2 PA10-6C Roboterarm und BarrettHand Dreifingerhand

### PA10-6C Roboterarm

Der Roboterarm PA10-6C ist von der Firma Mitsubishi Heavy Industries und besitzt 6 Freiheitsgrade. Der Roboterarm besteht aus 6 Komponenten. Von den 6 Komponenten sind drei Komponenten Drehgelenke und die anderen drei Komponenten sind Schwenkgelenke. In Tabelle 3.1 stehen die einzelnen Eigenschaften jedes Gelenkes des PA10-6C Roboterarmes, und Tabelle 3.2 enthält Geschwindigkeitsangaben der einzelnen Armkomponenten. Die Gesamtlänge des Roboterarmes beträgt 1317 mm und wiegt 38 kg. Die maximale Traglast ist mit 10 kg angegeben. Der Aufbau des Roboterarmes ist mit dem menschlichen Arm vergleichbar, siehe Abbildung 3.4. Der Controller des PA10 verfügt über eine ArcNet-Schnittstelle und kann von einem Standard-PC mit ArcNet-Interface gesteuert werden. Als Basis für die Steuerungssoftware wird RCCL (Robot-Control-C-Language) eingesetzt [34]. In Kapitel 2.2 gibt es eine verallgemeinerte Beschreibung vom kinematischen Aufbau eines Roboterarmes mit seinen Komponenten.

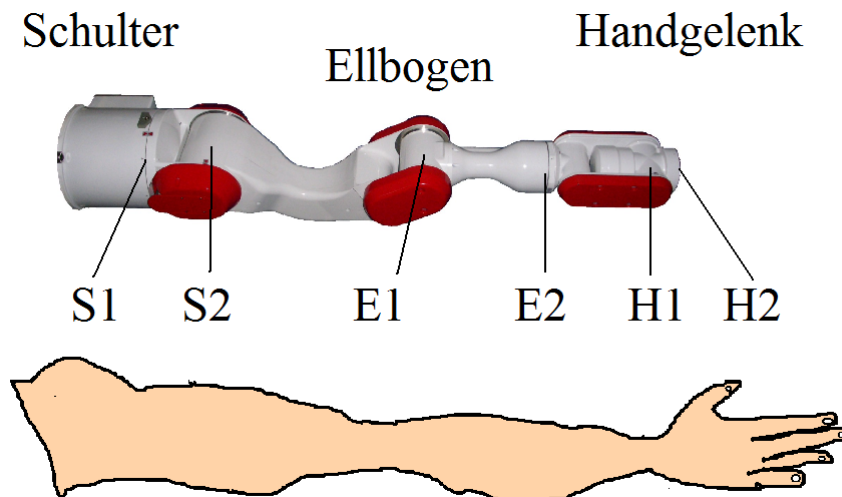


Abbildung 3.4: Roboterarm PA10-6C und Menschenarm.

Gelenkname	Mechanische Grenze	Controller Grenze	Software Grenze
S1 (Dreh)	$\pm 180^\circ$	$\pm 178^\circ$	$\pm 177^\circ$
S2 (Schwenk)	+ 127°, - 67°	$\pm + 125^\circ, -65^\circ$	+ 124°, - 64°
E1 (Dreh)	+ 164°, - 113°	+ 159°, - 108°	+ 158°, - 107°
E2 (Schwenk)	$\pm 270^\circ$	$\pm 256^\circ$	$\pm 255^\circ$
H1(Dreh)	$\pm 180^\circ$	$\pm 166^\circ$	$\pm 165^\circ$
H2 (Schwenk)	$\pm 270^\circ$	$\pm 256^\circ$	$\pm 255^\circ$

Tabelle 3.1: Eigenschaften der Gelenkkomponenten des PA10-6C.

Gelenkname	maximale Geschwindigkeit in rad/sek
S1	$\pm 1$
S2	$\pm 1$
E1	$\pm 2$
E2	$\pm 2\pi$
H1	$\pm 2\pi$
H2	$\pm 2\pi$

Tabelle 3.2: Geschwindigkeitsangaben der einzelnen Roboterarmkomponenten PA10-6C.

## BarrettHand Dreifingerhand

An dem Roboterarm PA10-6C ist die Dreifingerhand BarrettHand (Abbildung 3.5) montiert. Diese Dreifingerhand ist für Greifaktionen und Manipulation von Objekten gedacht. In dieser Arbeit wird diese Hand nur als Anbringungsrichtung für den Laserscanner genutzt.



Abbildung 3.5: BarrettHand.

### 3.3 HOKUYO URG-LX04



Abbildung 3.6: HOKUYO URG-LX04 2D Laserscanner.

In dieser Arbeit wird zur Datenerfassung der Tischszene der 2D Laserscanner HOKUYO URG-LX04 (Abbildung 3.6) der Firma SOKUIKI verwendet, der speziell für die Vermessung von Innenräumen entwickelt worden ist. Als Lichtquelle dient ein Infrarotlaser ( $\lambda = 765$ ). Der URG-LX04 misst die Abstände von Objekten durch das Phasennessverfahren Prinzip, siehe Kapitel 2.1.2. Er hat einen Öffnungswinkel von  $240^\circ$  und eine Winkelschrittweite von  $0.36^\circ$ . 682 Messwerte werden pro Scan gemessen. Ein Scan dauert in etwa 100 ms, 66 ms werden für die Vermessung und 33 ms werden für die Übersendung der Daten an den PC benötigt. Des Weiteren besitzt der Laserscanner zwei Schnittstellen, USB 2.0 und RS232C. In dieser Arbeit wird der Laserscanner über den USB Anschluss an den PC verbunden und benötigt eine externe Stromversorgung durch einen Steckverbinder. Die Daten werden über den USB versendet. Der HOKUYO ist klein, leicht und verbraucht wenig Strom.

Eigenschaften	Wert
Messbereich	20mm bis 4000mm
Scanzeit	100ms/Scan
Betriebsspannung	5V
Öffnungswinkel	$240^\circ$
Winkelschrittweite	$0.36^\circ$
Kommunikation	USB, RS-232 (seriell)
Protokoll	ScipVer1.0

Tabelle 3.3: Eigenschaften des HOKUYO URG-LX04.

In den zwei Abbildungen 3.7 und 3.8 sind die Scanebene und die Scanrichtung

des HOKUYO URG-04LX dargestellt. Weitere und ausführlichere Beschreibungen über den HOKUYO URG-LX04 sind in [35] und [36] zu finden.

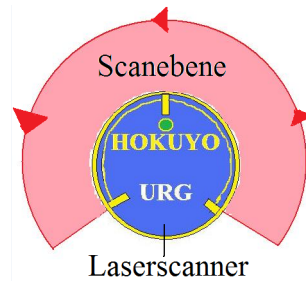


Abbildung 3.7: Hokuyo URG-04LX. Die Scanebene hat einen Öffnungswinkel von  $240^\circ$ .

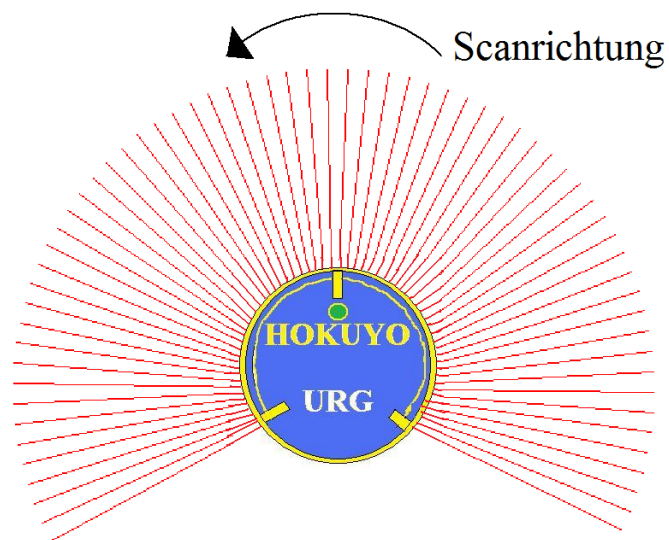


Abbildung 3.8: Hokuyo URG-04LX. Der Laserstrahl wird 682 mal entgegen dem Uhrzeigersinn in einem Winkelabstand von  $0.36^\circ$  ausgesandt.





# Kapitel 4

## Entwurf und Implementierungsdetails des URG-PA10 3D Lasersystems

In diesem Kapitel werden der Entwurf und einige Implementierungsdetails des URG-PA10 3D Lasersystems beschrieben. Das System basiert auf drei aufeinander bauenden Modulen, siehe Abbildung 4.1.

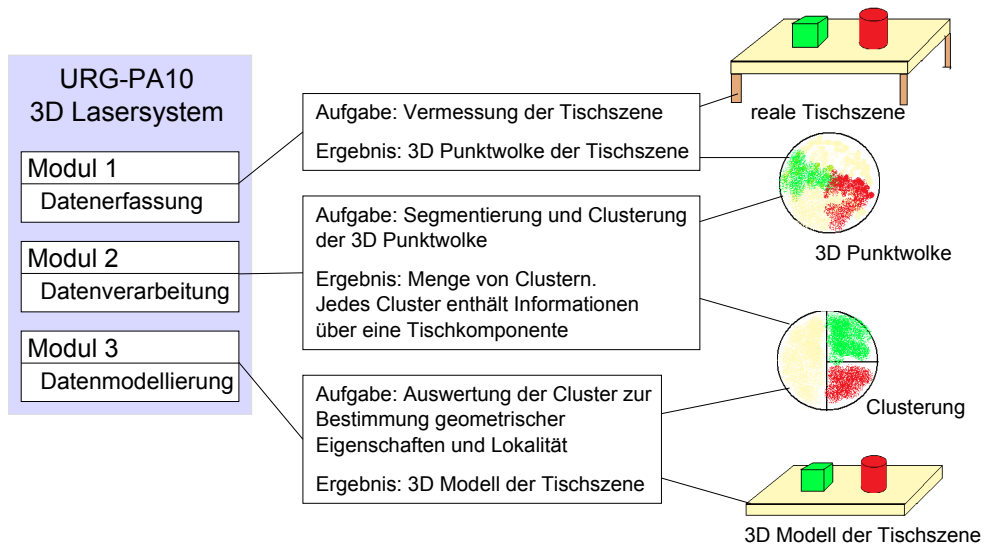


Abbildung 4.1: Entwurf für das URG-PA10 3D Lasersystem.

Auszüge aus diesem Kapitel sind in der Arbeit [37] publiziert.

## 4.1 Datenerfassung - Scanmethode (Modul 1)

Ziel der zu entwickelnden Scanmethode ist, eine 3D Punktwolke der Tischszene zu erzeugen. Die Hardwarekomponenten, die für die Scanmethode verwendet werden, sind

- der Hokuyo URG-LX04 2D Laserscanner und
- der PA10-6C Roboterarm des TASERs.

Technische Details über beide Komponenten sind in Kapitel 3 beschrieben. Beide Hardwarekomponenten sind miteinander nicht synchronisiert. In dieser Arbeit wird erstmals ein 2D Laserscanner an einen Roboterarm montiert, um 3D Daten einer Tischszene zu aquirieren. Vorteile des Roboterarmes sind, dass der Roboterarm aufgrund seiner kinematischen Eigenschaften als Dreh- und Schwenkvorrichtung und als mobile Trägerplattform mit einem eingegrenzten Bewegungsraum genutzt werden kann.

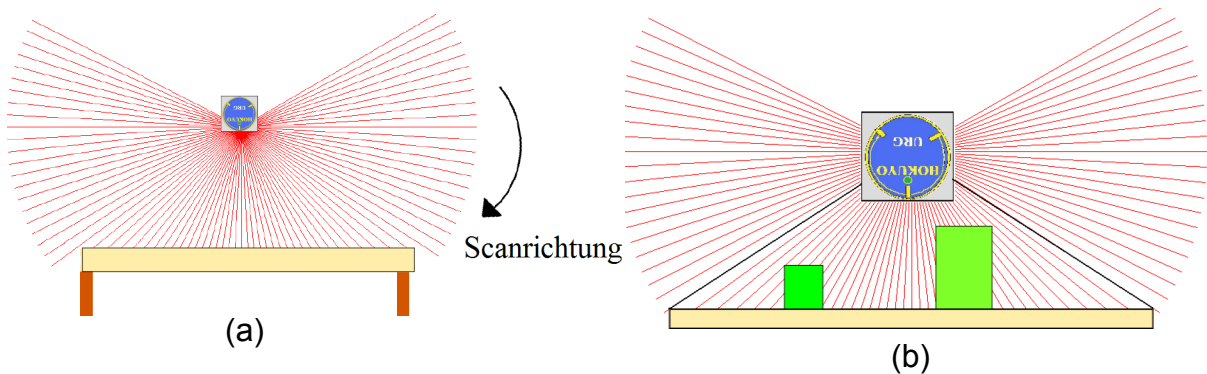


Abbildung 4.2: (a) Hokuyo URG-LX04 Scanebene liegt vertikal zur Tischoberfläche. (b) Eingegrenzter Bereich der Scanebene umfasst Tischoberflächenpunkte und Objektoberflächenpunkte.

Ein sinnvoller 2D Scan einer Tischszene ist gegeben, wenn die Scanebene vertikal zur Tischoberfläche, wie in Abbildung 4.2, liegt; denn Tisch- und

Objektoberflächenpunkte werden erfasst. Da nicht alle Messdaten eines 2D Scans der Tischszene zugehören, wird der Scanbereich des 2D Scans auf die Messdaten, die zur Tischszene gehören, eingegrenzt. Abbildung 4.3 zeigt, dass die Punktedichte eines 2D Scans von dem Abstand zwischen Laserscanner und Tischoberfläche abhängt. Je kleiner der Abstand ist, desto mehr Messwerte liegen von der Tischszene vor und desto dichter liegen die Punkte beieinander. Das Problem eines zu kleinen Abstandes ist, dass Objekte außerhalb des Scanbereiches liegen, siehe Abbildung 4.4 (b).

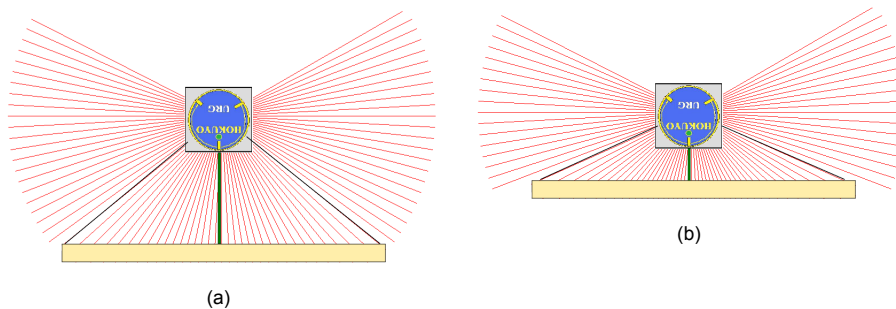


Abbildung 4.3: Unterschiedliche Abstandsgrößen zwischen Laserscanner und Tischoberfläche.

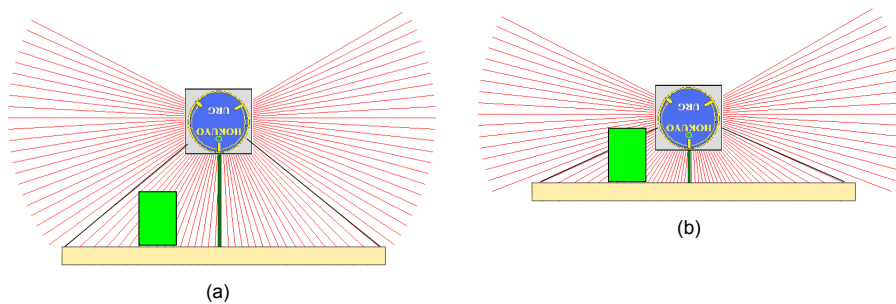


Abbildung 4.4: Objekt liegt (a) im Scanbereich. (b) außerhalb des Scanbereiches.

Mittels dem Roboterarm kann die Scanebene rotationell oder linear, wie in Abbildung 4.5, bewegt werden, um 3D Daten der Tischszene zu aquirieren. Die dargestellte rotationelle Scanmethode liefert eine 3D Punktwolke, deren höchste Punktedichte um die Drehachse herum ist. Die Punktedichte nimmt weiter weg von der Achse ab, und je weiter entfernt ein Objekt von der Drehachse liegt, desto weniger Messdaten liegen von dem Objekt vor, siehe Abbildung 4.6.

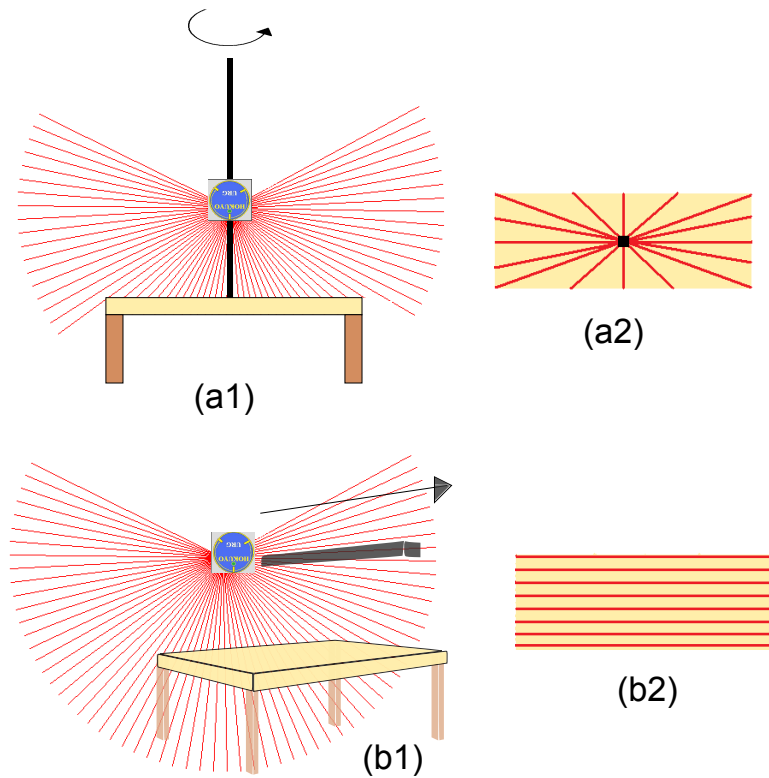


Abbildung 4.5: (a1) Rotationelle Scanmethode. (a2) Frontalsicht von oben auf die Tischfläche, 2D Scans (rote Linien). (b1) Lineare Scanmethode. (b2) Frontalsicht von oben auf die Tischfläche, 2D Scans (rote Linien).

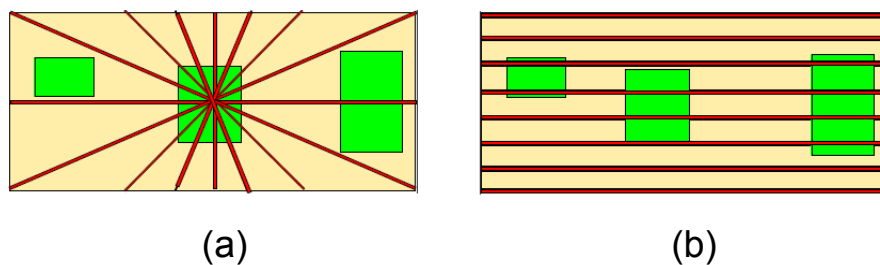


Abbildung 4.6: (a) Rotationelle Scanmethode. (b) Lineare Scanmethode.

Die lineare Scanmethode ähnelt dem Prinzip des Airborne Laserscanning (siehe Kapitel 2.1.3) und ist ein einfaches Messverfahren. Im Vergleich zu der rotationellen Methode erzeugt eine lineare Methode eine 3D Punktwolke der Tischszenen mit einer relativ gleichmäßigeren Punkteverteilung. Außerdem ist die

Eingrenzung des Scanbereiches für jeden 2D Scan gleich (siehe Abbildung 4.6). Für das URG-PA10 3D Lasersystem wird eine lineare Scanmethode entwickelt.

Folgende Punkte sind bei der zu entwickelnden linearen Scanmethode einzuhalten, die im Abschnitt 4.2 für die Clusteranalyse als Grundwissen über die 3D Punktwolke genutzt werden.

1. Alle Objekte müssen innerhalb des Scanbereiches liegen.
2. Objekte müssen eine Mindestgröße haben. Zu kleine Objekte werden kaum von Laserstrahlen erfasst (Abbildung 4.7 (b)).
3. Objekte dürfen nicht übereinanderliegen (Abbildung 4.7 (c)).
4. Objekte müssen einen Mindestabstand zueinander haben, sodass der Laserscanner einige Messdaten der Tischoberfläche erfassen kann, die zwischen den beiden Objekten liegen (Abbildung 4.7 (d)).
5. Mindestens die ersten und letzten drei Punkte jedes 2D Scans müssen Punkte der Tischoberfläche sein.

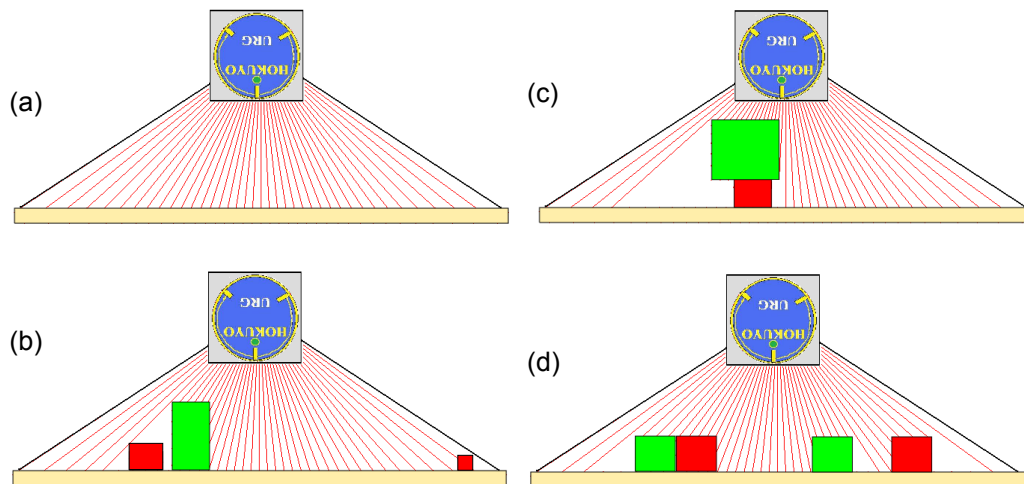


Abbildung 4.7: (a) Eingegrenzter Scanbereich. (b) Große Objekte verdecken kleine Objekte. (c) Objekte liegen übereinander. (d) Objekte liegen zu dicht nebeneinander.

### 4.1.1 Entwurf der linearen Scanmethode

Für die Realisierung der linearen Scanmethode für das URG-PA10 3D Lasersystem wird eine Armbewegung festgelegt, die den Laserscanner entlang einer Geraden führt. Die Gerade wird als Bewegungsbahn bezeichnet. Der Laserscanner wird an den Roboterarm angebracht, sodass die Scanebene bei der Durchführung der vordefinierten Armbewegung vertikal zur Tischoberfläche liegt, wie in Abbildung 4.8 dargestellt. Die Bewegungsbahn hat eine Länge von bis zu 460 mm. Der Startpunkt der Bewegungsbahn stellt den Koordinatenursprung eines dreidimensionalen kartesischen Koordinatensystems dar. Die Bewegungsbahn liegt auf der z-Achse. Dieses Koordinatensystem wird für die Bestimmung der Laserscannerposition und der Transformation der Laserdaten in kartesische Daten benutzt. Weitere Ansichten zur Betrachtung der linearen Scanmethode sind in den Abbildungen 4.9 und 4.10 zu sehen.

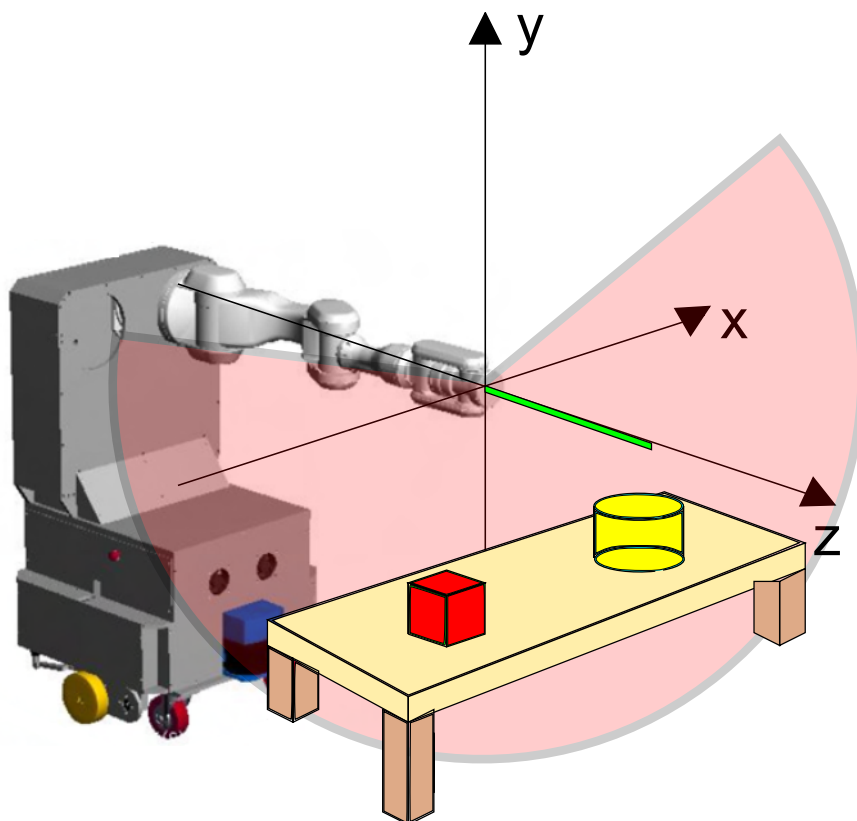


Abbildung 4.8: Der PA10-6C führt den 2D Laserscanner entlang der Bewegungsbahn (grüner Balken), die auf der z-Achse liegt.

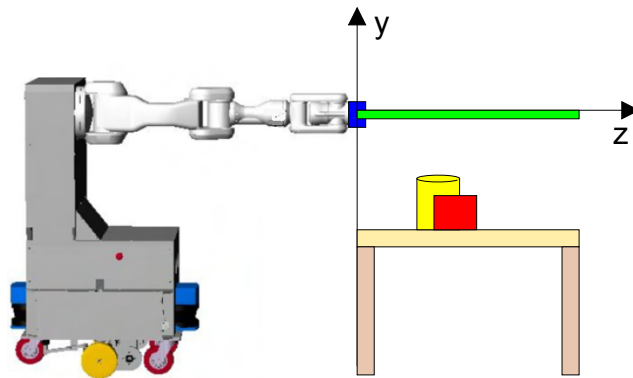


Abbildung 4.9: Seitliche Sicht von der Scanszene mit Roboterarm, Tischszene und Koordinatensystem.

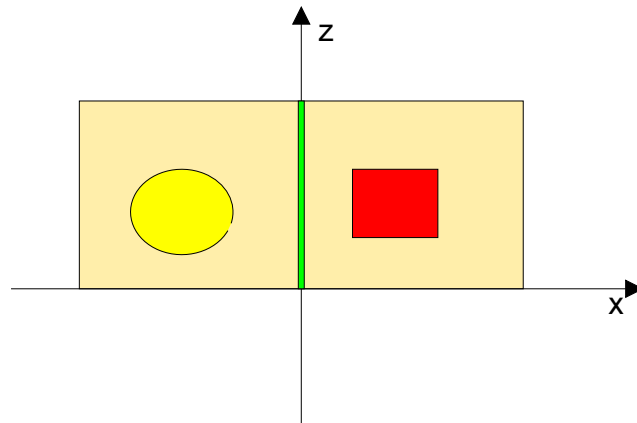


Abbildung 4.10: Frontalsicht von oben auf die zu vermessende Tischszene.

In dieser Arbeit wird die lineare Scanmethode auf zwei unterschiedlichen Arten implementiert:

- **Scan-Mit-Stop** Die Bewegungsbahn ist vom Startpunkt bis zum Endpunkt in äquidistante Stützpunkte unterteilt, siehe Abbildung 4.11 (b). Der Roboterarm wird so konfiguriert, dass er den Laserscanner von einem Stützpunkt zum nächsten führt. Bei jedem Stützpunkt hält der Roboterarm an, sodass der Laserscanner aus einer ruhigen Lage einen Scan durchführt und die Daten an den PC sendet. Zu jedem Scan wird die Position des Laserscanners durch die entsprechende Stützstelle zugeordnet.

- Scan-Ohne-Stop** Der Roboterarm durchläuft die Bewegungsbahn mit einer konstanten Geschwindigkeit ohne zu stoppen, siehe Abbildung 4.11 (c). Während der kontinuierlichen Armbewegung werden ständig Messdaten vom Laserscanner angefordert. Die Messdauer ist kürzer als bei der Scan-Mit-Stop Methode, da der Arm keine Stops macht. Das Problem dieser Methode ist, dass zu jedem Scan die Position des Laserscanners approximiert werden muss.

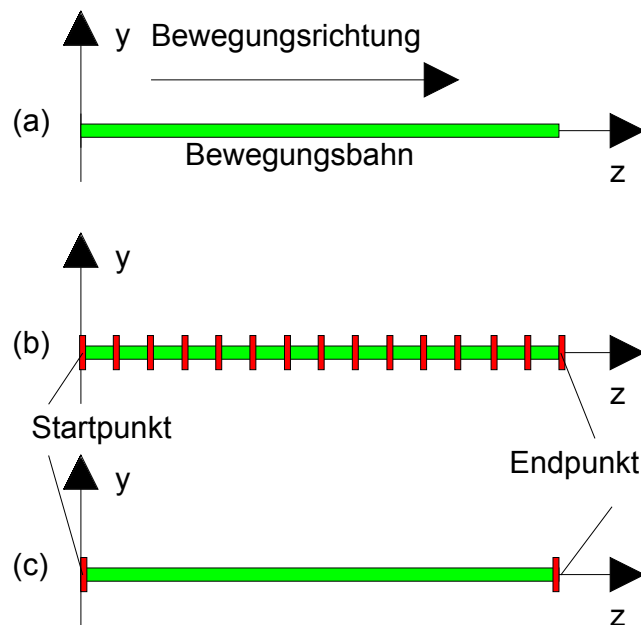


Abbildung 4.11: (a) Bewegungsbahn. (b) Scan-Mit-Stop. (c) Scan-Ohne-Stop.

Die Punktwolkendichte bei der Scan-Mit-Stop Methode hängt von dem gewählten Abstand ab, der die Bewegungsbahn in äquidistante Abschnitte unterteilt. Je kleiner der Abstand ist, desto dichter ist die 3D Punktwolke. Bei der Scan-Ohne-Stop Methode ist die Armgeschwindigkeit der entscheidende Faktor, der die Dichte der Punktwolke beeinflusst. Je langsamer sich der Arm entlang der Bewegungsbahn bewegt, desto mehr Scans erfolgen und führen zu einer dichteren 3D Punktwolke. Tabelle 4.1 zeigt die Vor- und Nachteile der Scanmethoden.



	Scan-Mit-Stop	Scan-Ohne-Stop
Vorteil	Position des Laserscanners ist zu jedem Scan bekannt	kurze Messdauer
Nachteil	lange Messdauer	Approximation der Laserscannerposition zu jedem Scan

Tabelle 4.1: Vor- und Nachteile der Scanmethoden.

## Transformation der Laserdaten in ein dreidimensionales kartesisches Koordinatensystem

Jeder 2D Scan ist eine Menge von Polarkoordinaten (siehe Kapitel 2.3.1). Zur Transformation der gesamten 2D Scans in ein dreidimensionales kartesisches Koordinatensystem wird das Zylinderkoordinatensystem aus Kapitel 2.3.1.1 verwendet. Die x und y Koordinaten für jeden Scan werden mit

$$x = R * \cos(\theta) \quad (4.1)$$

$$y = R * \sin(\theta) \quad (4.2)$$

berechnet.

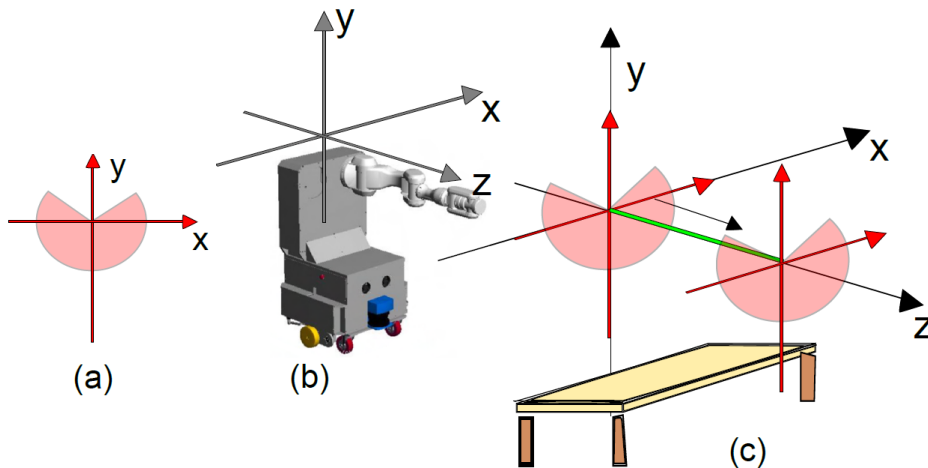


Abbildung 4.12: (a) Polarkoordinatensystem der Scanebene. (b) Roboterkoordinatensystem. (c) Verschiebung der Scanebene im 3D Raum.

Für alle Messdaten eines 2D Scans wird ein z-Koordinatenwert zugeordnet. Bei der Scan-Mit-Stop Methode wird für jeden 2D Scan der z-Wert aus den Stützpunkten entnommen. Bei der Scan-Ohne-Stop Methode werden die z-Werte durch eine Approximationsmethode bestimmt, indem mit Hilfe der Scananzahl ein Abstandswert berechnet wird, der die Bewegungsbahn vom Startpunkt bis zum Endpunkt in äquidistante Stützpunkte unterteilt. Die Anzahl der Stützpunkte ist gleich der Scananzahl. Jeder 2D Scan wird einem Stützpunkt zugeordnet, sodass der z-Wert ermittelt wird.

### Berechnung des Abstandes

$$a = \frac{s}{n - 1} \quad (4.3)$$

s ist die Länge der Bewegungsbahn.

n ist die Scananzahl

a ist der zu berechnende Abstand.

**Beispiel** (siehe zugehörige Abbildung 4.13)

s = 100 mm

n = 11

$$a = \frac{100\text{mm}}{11 - 1} = \frac{100\text{mm}}{10} = 10\text{mm}$$

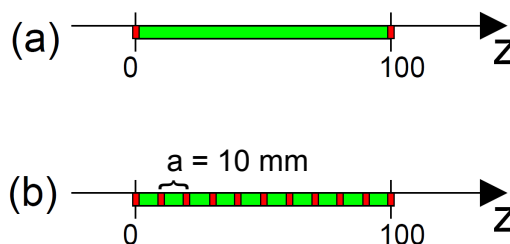


Abbildung 4.13: (a) Bewegungsbahn der Scan-Ohne-Stop Methode. (b) Unterteilung der Bewegungsbahn durch den errechneten Abstand 10 mm in 11 Stützstellen.

## 4.2 Clusteranalyse zur Segmentierung der 3D Punktwolke (Modul 2)

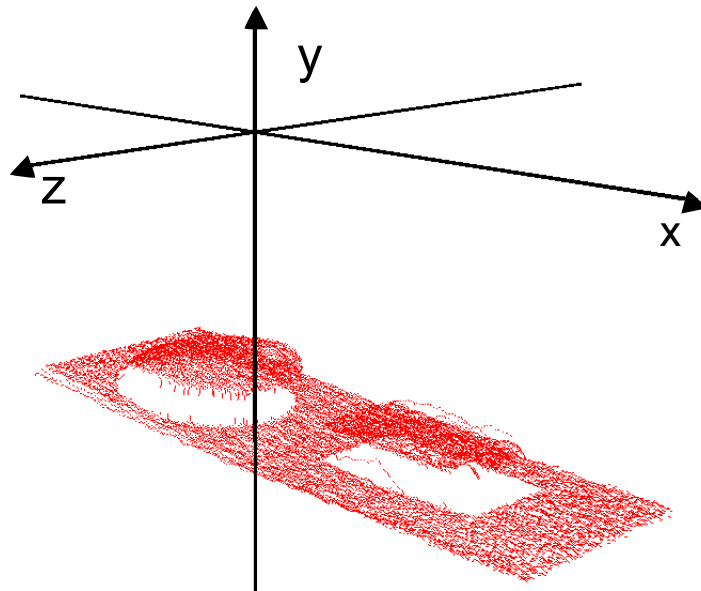


Abbildung 4.14: 3D Punktwolke der linearen Scanmethode.

In diesem Abschnitt wird ein dreistufiges Clusterverfahren vorgestellt. Ziel dieses Verfahren ist die 3D Punktwolke, die von der linearen Scanmethode aus Abschnitt 4.1.1 erzeugt wird, in Cluster zu segmentieren. Eins von den Clustern enthält Punkte, die zur Tischfläche gehören. Die anderen Cluster enthalten Punkte, die zu je einer Objektoberfläche der Tischszene gehören. Für dieses dreistufige Cluster-Verfahren werden Schwellwerte gesetzt, die zur Segmentierung und Sortierung der 3D Punkte benötigt werden. Das Diagramm aus Abbildung 4.15 zeigt den Ablauf der Clusteranalyse und die Anzahl der benötigten Schwellwerte für jede Cluster-Stufe.

Die 3D Punktwolke kann in folgender Mengenschreibweise beschrieben werden,

$$P = \{p_{00}, \dots, p_{0n}, \dots, p_{m0}, \dots, p_{mn}\} \quad (4.4)$$

mit  $p_{ij} \in \mathbb{R}^3$ ,  $m, n \in \mathbb{N}$

$m + 1$  entspricht der Scananzahl.

$n + 1$  ist die Anzahl der Punkte aus dem Scanbereich.

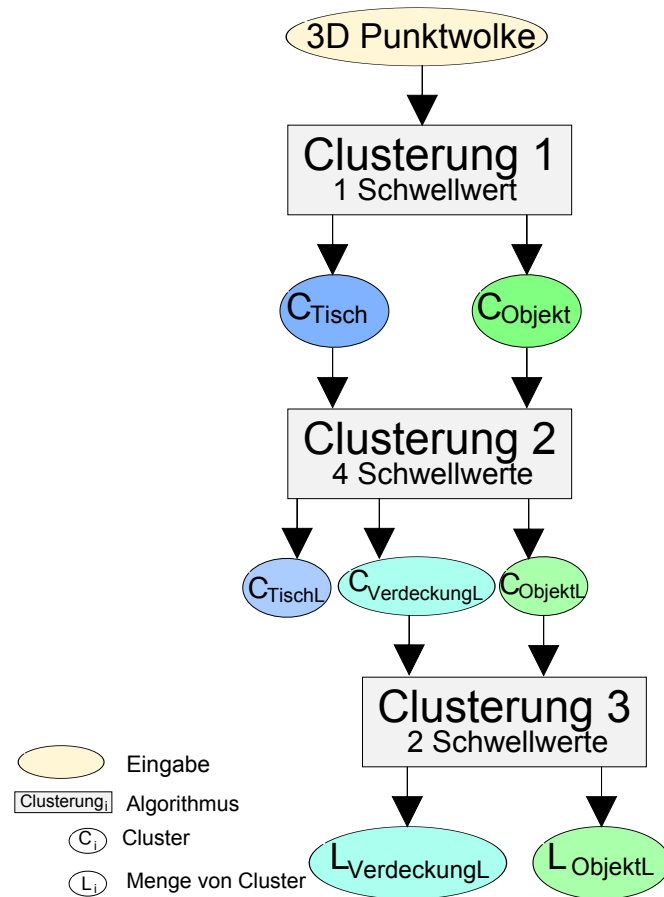


Abbildung 4.15: Diagramm des dreistufigen Clusterverfahrens.

Eine andere Schreibweise für diese 3D Punktwolke ist eine  $(m + 1) \times (n + 1)$  Matrix.

$$P = \begin{pmatrix} p_{00} & \dots & p_{0n} \\ p_{10} & \dots & p_{1n} \\ \vdots & \ddots & \vdots \\ p_{m0} & \dots & p_{mn} \end{pmatrix} \quad (4.5)$$

Diese  $(m + 1) \times (n + 1)$  Matrix wird als Eingabe für das dreistufige Clusterverfahren verwendet. Jede Zeile der Matrix entspricht den Messdaten eines 2D Scans. Alle Punkte einer Zeile haben den gleichen z-Koordinatenwert.

$$\exists_i \forall_{j=0}^m (\exists_z z_{ij} = z), p_{ij} = (x_{ij}, y_{ij}, z_{ij}) \quad (4.6)$$

Jede Zeile wird im weiteren Verlauf dieser Arbeit als Scanreihe bezeichnet.

### Clusterung 1

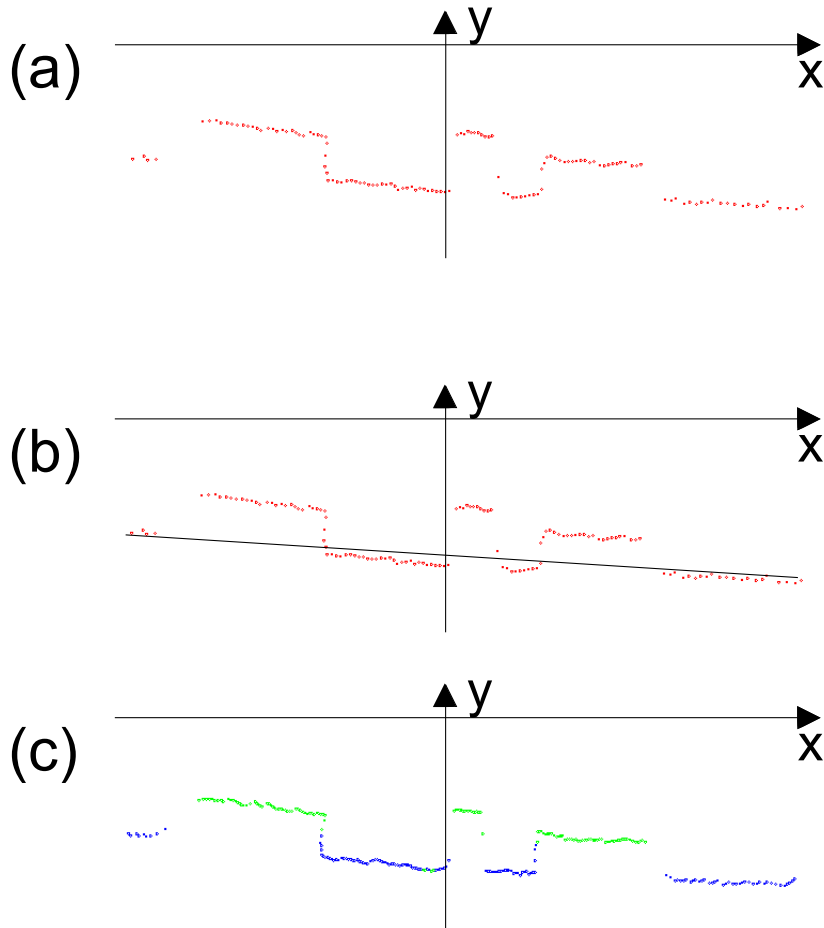


Abbildung 4.16: (a) Scanreihe. (b) Gerade zwischen dem ersten und letzten Punkt. (c) Blaue Punkte sind Tischoberflächenpunkte, grüne Punkte sind Objektoberflächenpunkte.

Im ersten Schritt des dreistufigen Clusterverfahrens wird eine disjunkte und exhaustive Clusterung durchgeführt, die die gesamten Punkte der 3D Punktwolke in genau zwei Cluster,  $C_{Tisch}$  und  $C_{Objekt}$ , segmentiert. Dem Cluster  $C_{Tisch}$  werden die Punkte zugeordnet, die einem Tischoberflächenpunkt entsprechen. Die anderen Punkte werden dem Cluster  $C_{Objekt}$  zugewiesen und werden als Objektoberflächenpunkte identifiziert.

Die Clusterung bearbeitet die Eingangsmatrix zeilenweise, indem zwischen dem

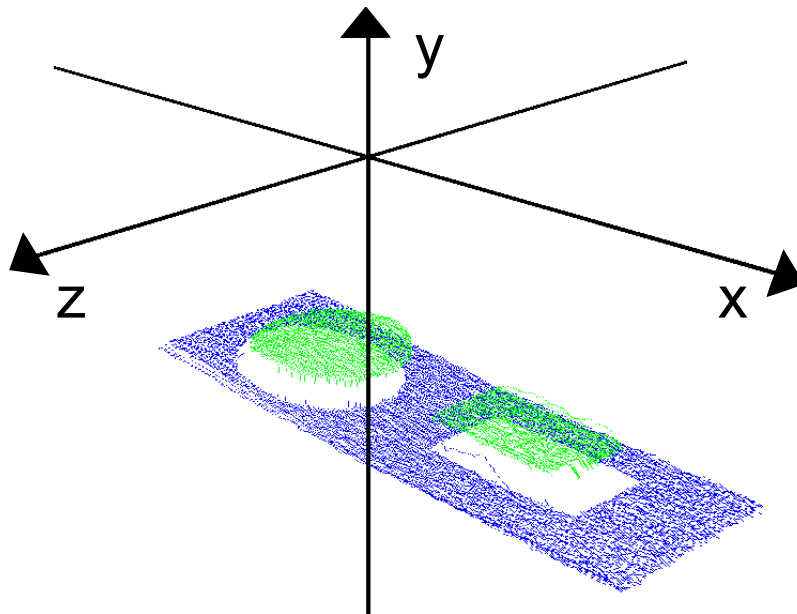


Abbildung 4.17: Ergebnis der ersten Clusterung.

ersten und letzten Punkt einer Scanreihe eine Gerade gezogen wird, wie in Abbildung 4.16 (b). Mit der Distanzfunktion auf Seite 22 wird der kürzeste Abstand zwischen jedem Punkt einer Scanreihe zu der zugehörigen Geraden berechnet. Ein konstant bleibender Schwellwert wird gesetzt. Punkte, deren Abstand größer als der gesetzte Schwellwert sind, werden dem Cluster  $C_{Objekt}$  eingeordnet. Punkte, deren Abstand zur Geraden kleiner als der gesetzte Schwellwert ist, werden dem Cluster  $C_{Tisch}$  zugeordnet.

In Abbildung 4.17 sieht man das Ergebnis der ersten Clusterung angewandt auf die 3D Punktwolke aus Abbildung 4.14. Die erste Clusterung  $L_1$  liefert die zwei Cluster  $C_{Tisch}$  und  $C_{Objekt}$ .

$$L_1 = \{C_{Tisch}, C_{Objekt}\} \quad (4.7)$$

## Clusterung 2

Die zweite Stufe des Clusterverfahrens bearbeitet die Cluster  $C_{Tisch}$  und  $C_{Objekt}$  getrennt voneinander. Die Hauptidee der zweiten Clusterung ist, eine Reihe von Punkten einer Scanreihe zu detektieren, die auf einer Linie liegen, siehe Abbildung 4.18.

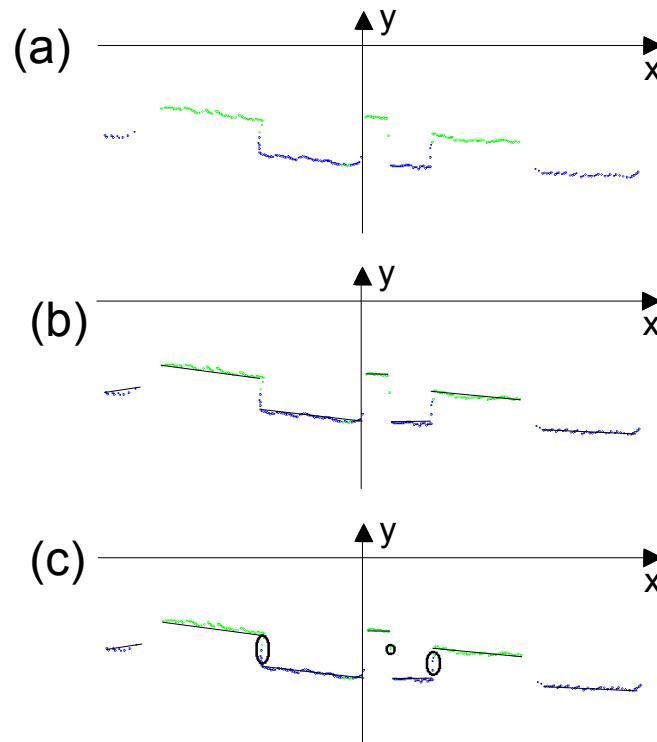


Abbildung 4.18: (a) Punkte aus den Clustern  $C_{Tisch}$  und  $C_{Objekt}$ . (b) Gruppierung von Punkten, die nah beieinander liegen. (c) Erster und letzter Punkt jeder Gruppe werden als Linie in das jeweilige Cluster  $C_{TischL}$  und  $C_{ObjektL}$  abgespeichert. Einige Punkte fallen weg (siehe eingekreiste Punktegruppe).

Es werden vier konstante Schwellwerte für dieses Clusterverfahren verwendet.

Der erste Schwellwert bestimmt den maximalen x-Abstand, den zwei aufeinanderfolgende Punkte einer Scanreihe haben dürfen. Der zweite Schwellwert bestimmt die maximale y-Differenz, die innerhalb einer Gruppe zwischen zwei Punkten vorliegen darf. Aus diesen beiden Schwellwertbedingungen werden Reihen von Punkten zu Gruppen zusammengefasst, sodass immer der erste und letzte Punkt jeder Gruppe als

Linie in die Cluster  $C_{TischL}$  bzw.  $C_{ObjektL}$  eingefügt werden. Die Punkte, die zwischen dem ersten und letzten Punkt jeder Gruppe liegen, werden für die Weiterverarbeitung nicht mehr genutzt. Ein dritter Schwellwert gibt eine Mindestanzahl vor, die entscheidet, wieviel Punkte eine Gruppe haben muss, um die Gruppe als Linie zu definieren. Liegt die Anzahl der Punkte einer Gruppe unter dieser Mindestanzahl, so fallen diese Punkte für die Weiterverarbeitung weg, siehe Abbildung 4.18 (c).

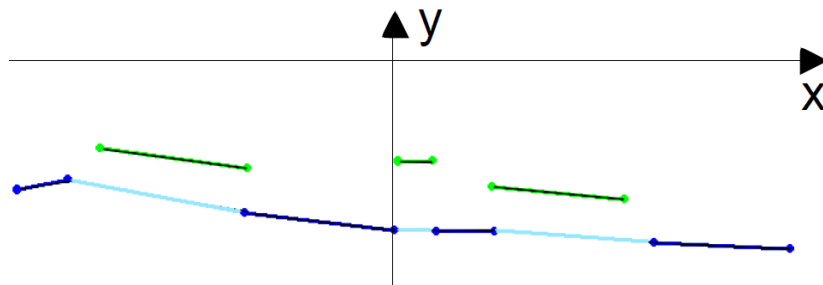


Abbildung 4.19: Verdeckungslinien (hellblau) werden aus zwei aufeinanderfolgenden Tischlinien erzeugt.

Der vierte Schwellwert wird dazu verwendet, um aus dem Cluster  $C_{TischL}$  einen weiteren Cluster  $C_{VerdeckungL}$  zu bilden. Die Entfernung zwei aufeinanderfolgenden Linien einer Scanreihe aus dem Cluster  $C_{TischL}$  wird berechnet. Überschreitet der Entfernungswert den vierten Schwellwert, so wird eine Linie aus den beiden Linien aus  $C_{TischL}$  bestimmt und in den Cluster  $C_{VerdeckungL}$  gelegt, siehe Abbildung 4.19.

Die drei Cluster  $C_{TischL}$ ,  $C_{ObjektL}$  und  $C_{VerdeckungL}$  sind in Abbildung 4.20 und 4.21 zu sehen.

Die zweite Clusterung  $L_2$  liefert die drei Cluster  $C_{TischL}$ ,  $C_{ObjektL}$  und  $C_{VerdeckungL}$ .

$$L_2 = \{C_{TischL}, C_{ObjektL}, C_{VerdeckungL}\} \quad (4.8)$$



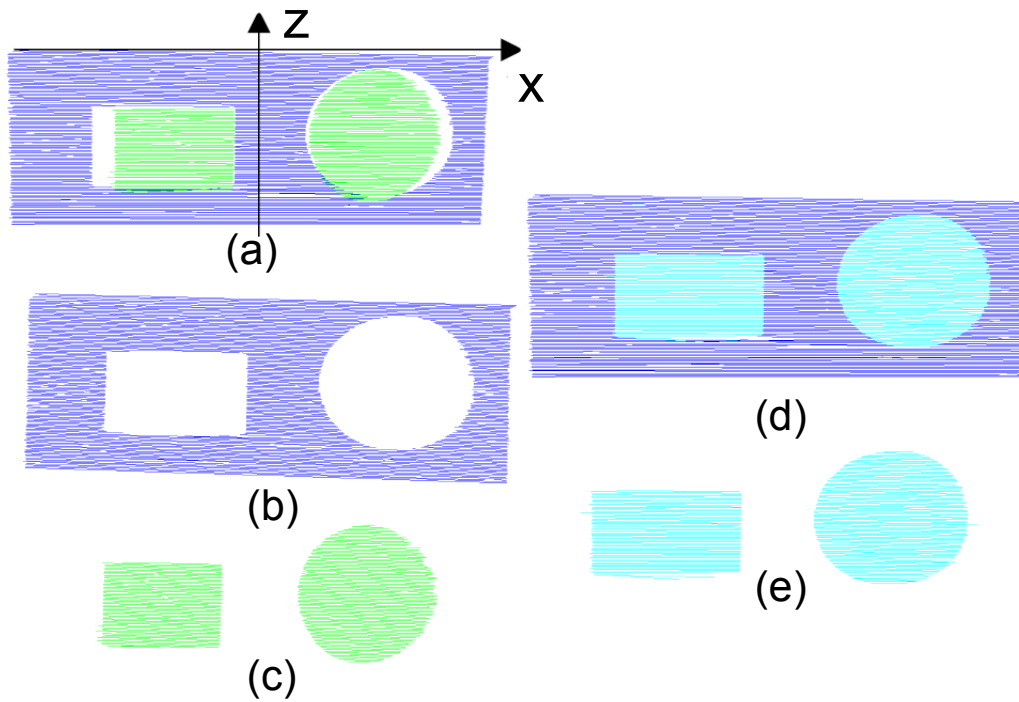


Abbildung 4.20: (a)  $C_{TischL}$  (blau),  $C_{ObjektL}$  (grün). (b)  $C_{TischL}$ . (c)  $C_{ObjektL}$ . (d)  $C_{TischL}$  (blau),  $C_{VerdeckungL}$  (hellblau). (e)  $C_{VerdeckungL}$ .

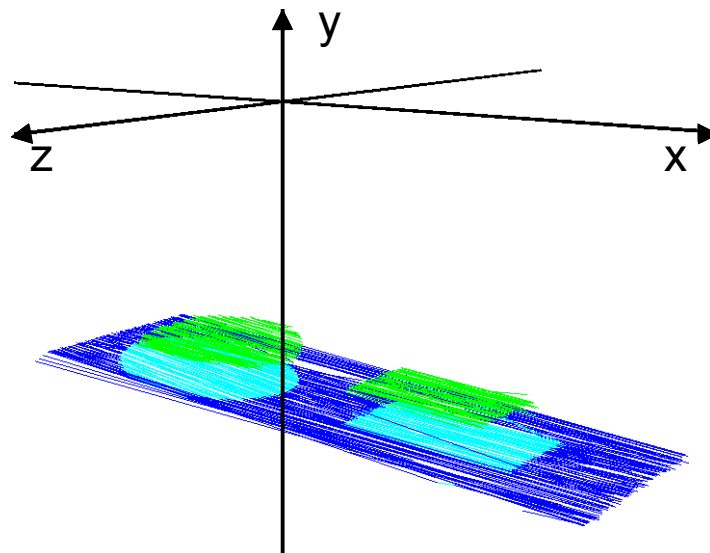


Abbildung 4.21: 3D Linienwolke der Cluster  $C_{TischL}$ ,  $C_{ObjektL}$  und  $C_{VerdeckungL}$ .

### Clusterung 3

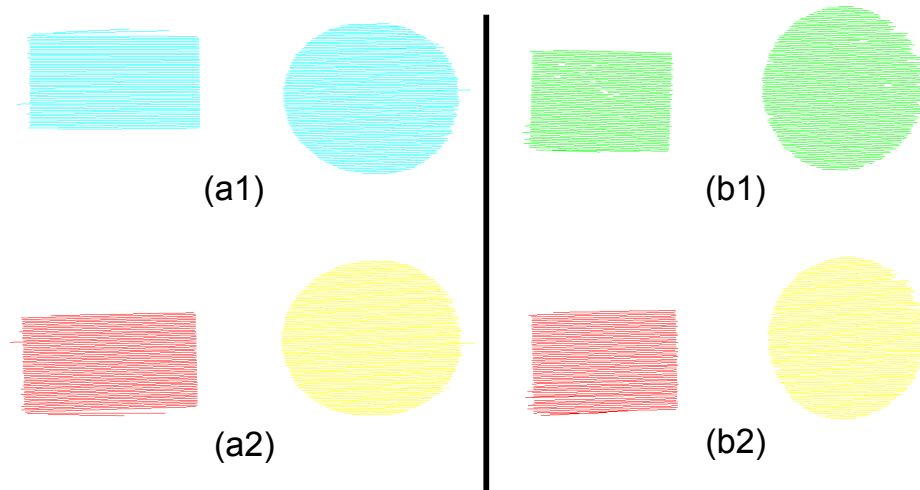


Abbildung 4.22: (a1) Unsortierte Verdeckungslinienwolke. (a2) Sortierte Verdeckungslinienwolke. (b1) Unsortierte Objektlinienwolke. (b2) Sortierte Objektlinienwolke.

Die dritte Clusterung wird auf die Cluster  $C_{ObjektL}$  und  $C_{VerdeckungL}$  getrennt angewandt. Beide Cluster enthalten eine unsortierte Menge an Linien (Linienwolke). Ziel der dritten Clusterung ist, die Linienwolke,  $C_{ObjektL}$  und  $C_{VerdeckungL}$ , in Cluster zu sortieren. Jedes erzeugte Cluster enthält Linien, die zu genau einem Objekt auf der Tischfläche gehören. Diese Clusterung ist disjunktiv und liefert eine unbekannte Anzahl von Clustern. Gewünscht ist, dass die Anzahl der entstandenen Cluster gleich der Anzahl der Objekte, die sich auf der realen Tischszene befinden, ist.

Der Algorithmus dieser dritten Stufe vergleicht jeweils zwei Linien, die aus zwei aufeinanderfolgenden Scanreihen sind. Vergleichskriterium ist der Abstand zwischen den Anfangs- und Endpunkten der zwei Linien. Liegen die Abstände der x-Werte der jeweiligen Anfangs- bzw. Endpunkten unterhalb eines gesetzten Schwellwertes, so wird ein Cluster erzeugt, dem die beiden Linien zugeordnet werden. Sollte sich eine der Linien schon in einem Cluster befinden, so wird die andere Linie ebenfalls dem Cluster zugeordnet. Für den Fall, dass beide Linien als nicht zusammengehörig identifiziert werden, fallen die beiden Linien in getrennte Cluster. Aufgrund von Messfehlern und Ungenauigkeiten kommt es

vor, dass viele Cluster nur ein bis zwei Linien enthalten. Daher wird ein Schwellwert gesetzt, der bestimmt, wieviele Linien ein Cluster haben muss, um in die Clusterung aufgenommen zu werden. Ein Ergebnis der dritten Clusterung ist in Abbildung 4.22 dargestellt.

Die dritte Clusterung liefert folgende zwei Clusterungen

$$L_{ObjektL} = \{C_0, \dots, C_s\} \quad (4.9)$$

$$L_{VerdeckungL} = \{C_0, \dots, C_t\} \quad (4.10)$$

mit  $s = t$ . Aufgrund von Messfehlern oder 3D Punktwolken, die eine geringe Punktedichte haben, kann  $s$  ungleich  $t$  sein.

In Abbildung 4.23 sind die sortierte Objektlinienwolke und die sortierte Verdeckungslinienwolke dargestellt. Jeder sortierten Objektlinienwolke liegt eine Linienwolke aus der sortierten Verdeckungslinienwolke vor.

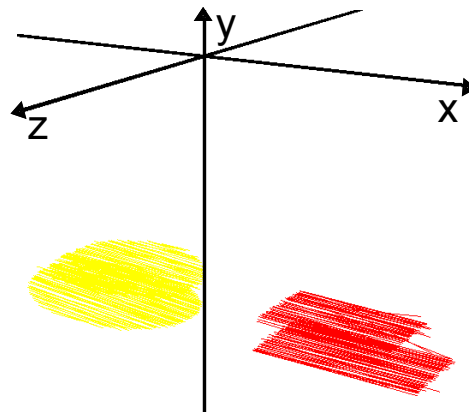


Abbildung 4.23: Sortierte Objektlinienwolke und Verdeckungslinienwolke.

Das Endergebnis des dreistufigen Clusterverfahrens sind die folgenden drei Cluster

- $C_{TischL}$ ,
- $L_{ObjektL}$  (Menge von Cluster) und
- $L_{VerdeckungL}$  (Menge von Cluster).

## 4.3 Wissensextraktion und Modellierung (Modul 3)

Bei der Wissensextraktion und Analyse wird ein gewisses Maß an Wissen über die Tischszene vorausgesetzt.

### 4.3.1 Auswertung der Tischlinienwolke

Die Tischoberfläche wird als rechteckförmig vorausgesetzt.

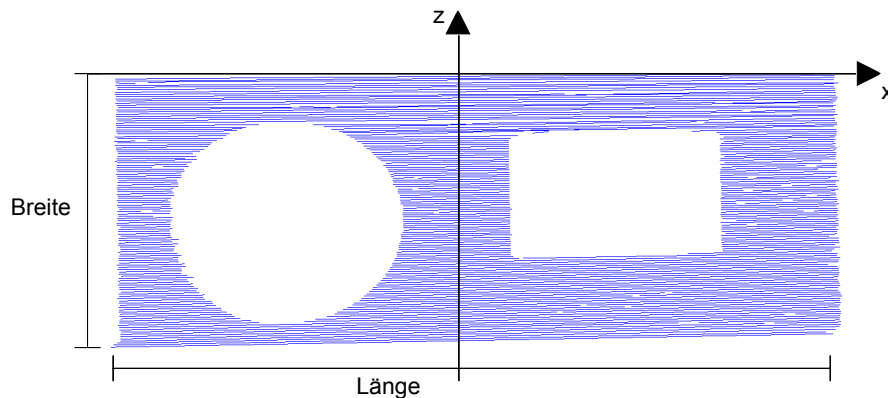


Abbildung 4.24: Tischlinienwolke  $C_{TischL}$ . Breite und Länge der Tischoberfläche.

Die **Breite** des Tisches ist aus dem z-Koordinatenwert der letzten Scanreihe ableitbar. Die erste Scanreihe hat als z-Koordinate immer den Wert Null, siehe Abbildung 4.24.

Die **Länge** wird durch den Abstand des ersten Punktes und des letzten Punktes der ersten Scanreihe bestimmt.

Die **Höhe** kann aus dem Cluster  $C_{TischL}$  nicht bestimmt werden, daher wird ein Wert gesetzt, der für alle Tischflächen gilt.

### 4.3.2 Analyse und Auswertung der markierten Objektlinienwolke und Verdeckungslinienwolke

Die Cluster  $L_{ObjektL}$  und  $L_{VerdeckungL}$  aus Abschnitt 4.2

$$L_{ObjektL} = \{CO_0, \dots, CO_s\} \quad (4.11)$$

$$L_{VerdeckungL} = \{CV_0, \dots, CV_s\} \quad (4.12)$$

mit  $\forall_i CO_i = (p_{a_i}, p_{e_i})$ ,  $\forall_i CV_i = (p_{a_i}, p_{e_i})$ ,

$0 \leq i \leq s$ ,  $i, s \in \mathbb{N}$

$p_{a_i}, p_{e_i} \in \mathbb{R}^3$

werden zu einer Objektmenge  $O$

$$O = \{(CO_0, CV_0), \dots, (CO_s, CV_s)\} \quad (4.13)$$

zusammengefasst.

Jedes Element  $e_i \in O$  ist ein zweier Tupel  $(CO_i, CV_i)$  und besteht aus einer Objektlinienwolke  $CO_i$  und einer Verdeckungslinienwolke  $CV_i$ , die zu genau einem Objekt der Tischszene gehören. Jedes Element  $e_i \in O$  wird einzeln analysiert, ausgewertet und modelliert und durchläuft die folgenden drei Schritte:

1. Bestimmung des Grundkörpers
2. Bestimmung der geometrischen Größen
3. Bestimmung der Lage im Raum

## 1. Bestimmung des Grundkörpers

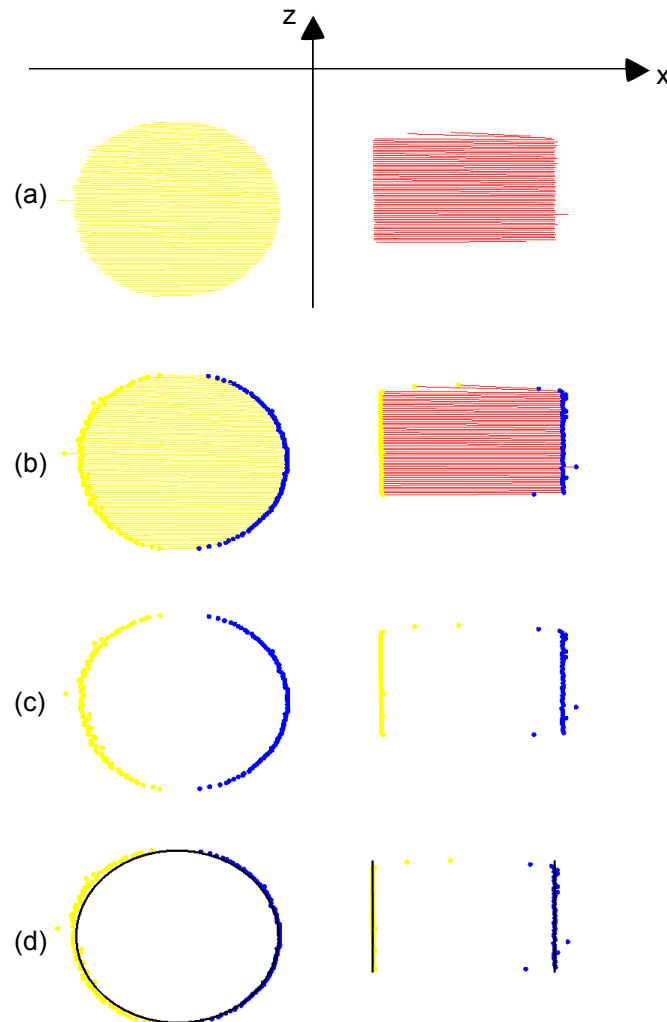


Abbildung 4.25: (a)  $L_{VerdeckungL}$ . (b) Anfangspunkte (blau) und Endpunkte (gelb) mit Linien. (c) Anfangs- und Endpunkte. (d) Ellipse und paralleles Geradenpaar.

Zur Bestimmung des Grundkörpers werden aus dem Element  $e_i$  die Verdeckungslinienwolke  $CV_i$  und die Objektlinienwolke  $CO_i$  entnommen. Die Linienwolken werden als Punktemenge betrachtet. Jede der beiden Punktemengen wird einzeln überprüft, ob die Anreihung der Punkte einer Ellipse oder einem parallelen Geradenpaar entspricht.

In Abbildung 4.25 (c) sind zwei Punktemengen von zwei Linienwolken dargestellt. Die Anreihung der Punkte der einen Punktemenge ähnelt einer

Ellipse und kann auf ein Zylinderobjekt zurückgeführt werden. Die andere Punktemenge enthält Punkte, deren Anreihung einem parallelen Geradenpaar entspricht, siehe Abbildung 4.25 (d). Ein paralleles Geradenpaar ist ein Hinweis darauf, dass es sich um ein Quader-Objekt handeln kann. Eine Ellipse und ein paralleles Geradenpaar sind zweidimensionale Quadriken (siehe Kapitel 2.4.3).

### Untersuchung der Punktemenge auf ein paralleles Geradenpaar

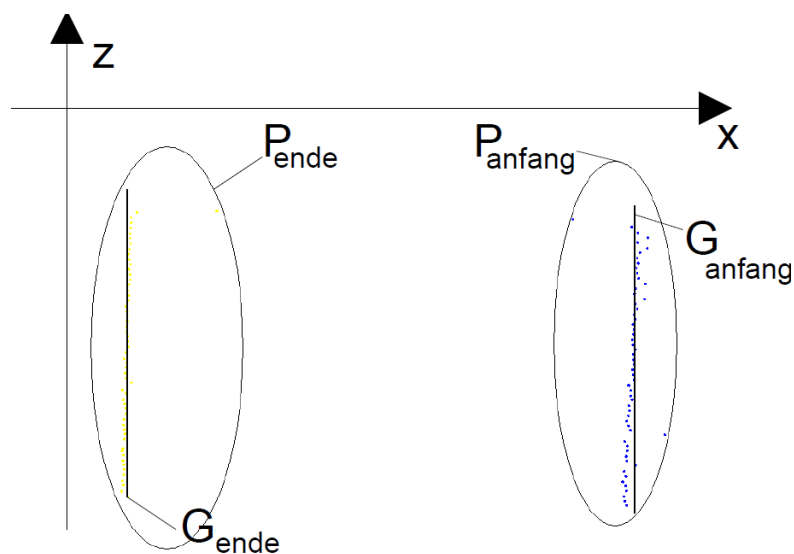


Abbildung 4.26: Punktemenge wird in Anfangs- und Endpunktemengen zur Überprüfung eines parallelen Geradenpaares zerlegt.

Für die Untersuchung, ob die Punktemenge einer Linienwolke einem parallelen Geradenpaar entspricht, wird die Linienwolke in die zwei folgenden Punktemengen

$$P_{anfang} = \{p_{a0}, \dots, p_{an}\} \quad (4.14)$$

$$P_{ende} = \{p_{e0}, \dots, p_{en}\} \quad (4.15)$$

zerlegt.

Die Menge  $P_{\text{anfang}}$  enthält alle Anfangspunkte der Linien und  $P_{\text{ende}}$  enthält alle Endpunkte der Linien. Zu jeder der beiden Punktmenge wird eine Gerade bestimmt, siehe Abbildung 4.26.

$$G_{\text{anfang}} = ((x_{aD}, 0, z_{a0}), (x_{aD}, 0, z_{an})) \quad (4.16)$$

$$G_{\text{ende}} = ((x_{eD}, 0, z_{e0}), (x_{eD}, 0, z_{en})) \quad (4.17)$$

Die x-Koordinate  $x_{aD}$  aus (4.16) bzw.  $x_{eD}$  aus (4.17) wird durch das arithmetische Mittel der x-Werte aus der Anfangspunktmenge bzw. Endpunktmenge berechnet.

Wenn die Geraden feststehen, wird der Abstand jedes Punktes zur entsprechenden Geraden berechnet, um zu überprüfen, ob die Punkte nahe oder auf der Geraden liegt. Der Abstand zwischen einem Punkt zu der Geraden erfolgt wie in Kapitel 2.17 beschrieben. Auswertungsergebnisse zeigen, dass wenn mindestens 80 Prozent der Punkte der Anfangspunktmenge bzw. Endpunktmenge nahe oder auf der Geraden sich befinden, die Punktmenge den Kriterien eines parallelen Geradenpaares erfüllt. Die übrigen 20 Prozent der Punkte sind auf Messfehler zurückzuführen.

## Untersuchung der Punktmenge auf eine Ellipse

Für die Untersuchung, ob die Anreihung der Punkte einer Ellipse ähnelt, wird als erster Schritt der Mittelpunkt  $P_M(x_M, z_M)$  der gesamten Punktmenge bestimmt, siehe Abbildung 4.27. Mit Hilfe dieses Mittelpunktes wird eine Translationsmatrix erstellt, um die Punkte zu verschieben. Ziel ist, dass die Punkte, wie in Abbildung 4.28, um den Koordinatenursprung liegen.

Translationsformel zur Verschiebung der Punktmenge

$$\begin{pmatrix} x' \\ z' \\ 1 \end{pmatrix} = \begin{pmatrix} x + x_M \\ z + z_M \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & x_M \\ 0 & 1 & z_M \\ 0 & 0 & 1 \end{pmatrix} * \begin{pmatrix} x \\ z \\ 1 \end{pmatrix} \quad (4.18)$$



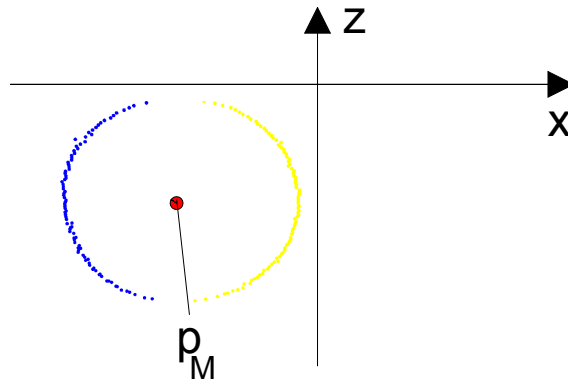


Abbildung 4.27: Bestimmung des Mittelpunktes.

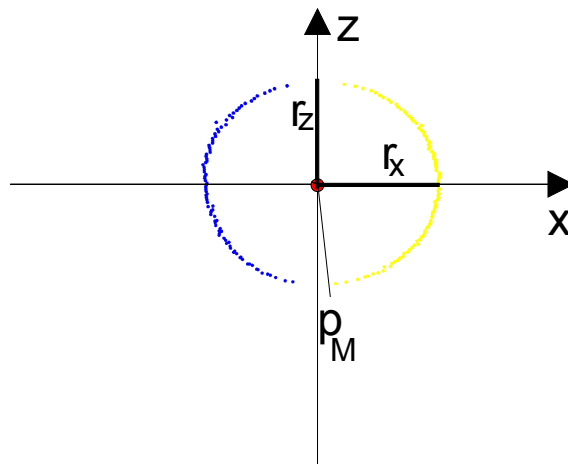


Abbildung 4.28: Transformierte Punkte. Bestimmung der Radien  $r_x$  und  $r_z$ .

Nach der Transformation werden die Radien  $r_x$  und  $r_z$  bestimmt. Im Idealfall erfüllt jeder transformierte Punkt  $p(x', y')$  die Gleichung (4.19).

$$-\frac{x'^2}{r_x^2} - \frac{z'^2}{r_z^2} + 1 = 0 \quad (4.19)$$

Eine Reihe von Messergebnissen zeigt, dass durch Messungenauigkeiten und Messfehler die Gleichung aus (4.19) in (4.20) abgeändert werden muss.

Mindestens 80 Prozent der transformierten Punkte muss die Gleichung aus (4.20) erfüllen, um den Kriterien einer Ellipse zu entsprechen.

$$-\frac{x'^2}{r_x^2} - \frac{z'^2}{r_z^2} + 1 \leq 0.02 \quad (4.20)$$

## Entscheidungsverfahren

Das Entscheidungsverfahren zur Festlegung der Oberfläche eines Elementes  $e_i \in O$  basiert auf den Ergebnissen der Untersuchungen der Punktemenge, die auf Seite 63 und 64 beschrieben sind. Die zwei Untersuchungen sind als Methoden, `Geraden()` und `Ellipse()`, implementiert. Die Linienwolken,  $CO_i$  und  $CV_i$ , eines Elementes  $e_i \in O$  werden als Parameter beiden Methoden übergeben. Jede Methode liefert einen booleschen Rückgabewert. Der Rückgabewert **false** bedeutet, dass die Linienwolke den Bedingungen eines parallelen Geradenpaars bzw. einer Ellipse nicht entspricht. Der Wert **true** hingegen bedeutet, dass die Bedingungen erfüllt sind. Tabelle 4.2 zeigt alle möglichen Ergebniskombinationen der zwei Methoden angewandt auf  $CO_i$  und  $CV_i$ . Die letzte Spalte der Tabelle zeigt die Entscheidung, ob ein Element  $e_i$  als Quader, Zylinder oder Polyeder modelliert wird. Ein Objekt, das als Polyeder modelliert wird (Abschnitt 4.3.3), wird als unbekannt deklariert.

<code>Geraden(<math>CO_i</math>)</code>	<code>Ellipse(<math>CO_i</math>)</code>	<code>Geraden(<math>CV_i</math>)</code>	<code>Ellipse(<math>CV_i</math>)</code>	<b>3D Modell</b>
false	false	false	false	Polyeder
false	false	false	true	Zylinder
false	false	true	false	Quader
false	false	true	true	Polyeder
false	true	false	false	Zylinder
false	true	false	true	Zylinder
false	true	true	false	Polyeder
false	true	true	true	Zylinder
true	false	false	false	Quader
true	false	false	true	Polyeder
true	false	true	false	Quader
true	false	true	true	Quader
true	true	false	false	Polyeder
true	true	false	true	Zylinder
true	true	true	false	Quader
true	true	true	true	Polyeder

Tabelle 4.2: Entscheidungstabelle zur Oberflächenmodellierung.

## Sonderfall

Wenn die Kanten eines Quaders nicht parallel zu den x- und z-Achsen verlaufen, wird dieser Quader als Sonderfall bezeichnet, siehe Abbildung 4.29.

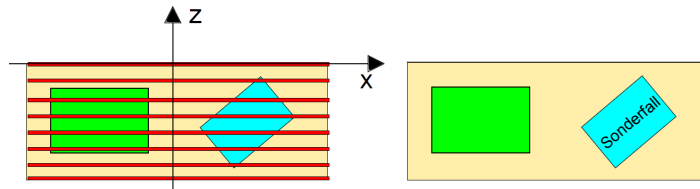


Abbildung 4.29: Die Kanten des blauen Quaders liegen nicht parallel zu den x- und z-Achsen und auch nicht zu den roten Scanlinien.

In Abbildung 4.30 (c) sind Anfangs- und Endpunkte dargestellt, die in dieser Arbeit einem Sonderfall zugeordnet werden. Die Anreihung der Punkte gleicht keinem parallelen Geradenpaar, wie in Abbildung 4.25 (d).

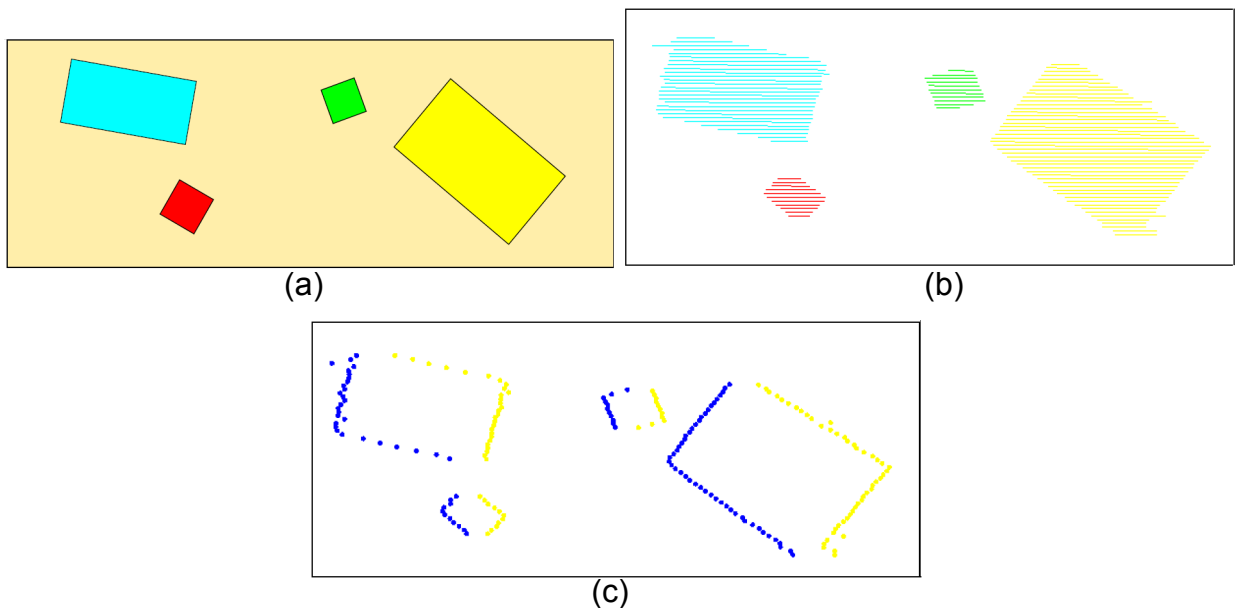


Abbildung 4.30: (a) Sonderfälle. (b) Linienwolke. (c) Anfangs- und Endpunkte.

Merkmal der jeweiligen Anfangs- und Endpunkte der Sonderfälle ist, dass sie auf zwei Geraden, die sich in einem Schnittpunkt treffen, dargestellt werden können. Ist der Schnittpunkt bekannt, so können die zwei Geraden vom

Schnittpunkt bis zum ersten bzw. letzten Punkt der Punktmenge gezogen werden. Danach kann überprüft werden, ob die dazwischenliegenden Punkte nahe oder auf der jeweiligen Geraden sich befinden. In Abbildung 4.31 sind zwei

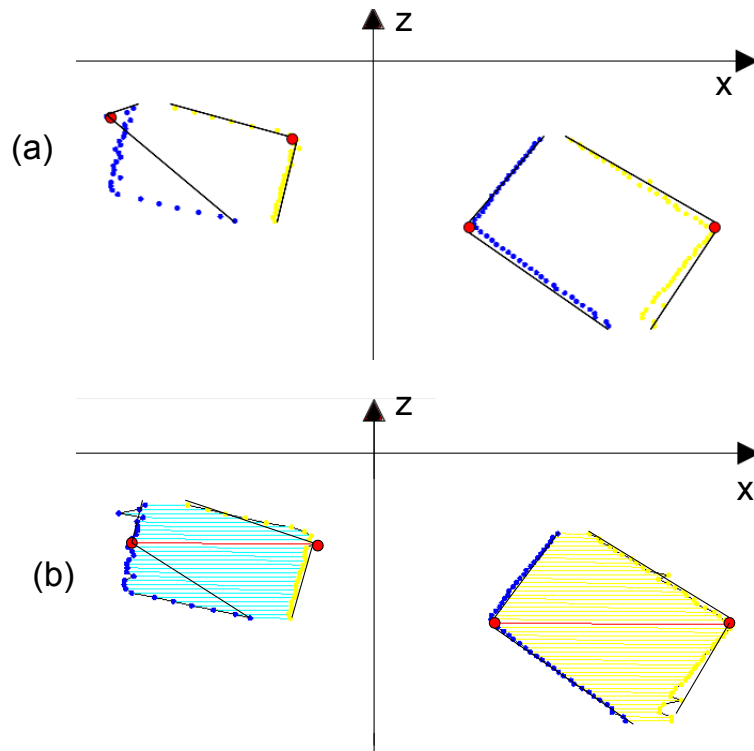


Abbildung 4.31: Lösungsansätze für die Analyse von Sonderfällen (a) Schnittpunkt wird durch den kleinsten bzw. größten  $x$ -Wert bestimmt. (b) Schnittpunkte werden durch die längste Linie (rot) festgelegt.

Lösungsansätze zur Bestimmung der Schnittpunkte dargestellt. Der erste Lösungsansatz markiert den Punkt mit dem größten  $x$ -Wert als Schnittpunkt für die Anfangspunktmenge.

Für die Endpunktmenge wird der Punkt mit dem kleinsten  $x$ -Wert als Schnittpunkt markiert. Das Problem bei diesem Lösungsansatz ist, dass ein falscher Schnittpunkt aufgrund von Messfehlern bestimmt wird, siehe Abbildung 4.31 (a). Ein anderer Ansatz, der in Abbildung 4.31 (b) zu sehen ist, basiert auf der längsten Linie der jeweiligen Linienwolken. Der Anfangs- und Endpunkt der längsten Linie werden als Schnittpunkte festgelegt. Das Problem bei diesem zweiten Lösungsansatz ist, dass die längste Linie für einige Sonderfälle nicht die gewünschten Schnittpunkte liefert. Eine Kombination aus

beiden Ansätzen könnte ein robusteres Untersuchungsverfahren erzeugen, das hier in dieser Arbeit nicht weiter behandelt wird.

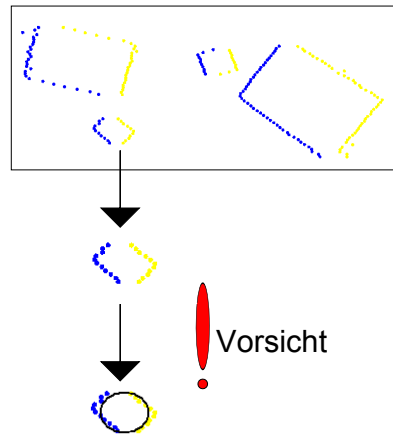


Abbildung 4.32: Quader-Objekte mit einer quadratischen Grundfläche laufen Gefahr, als Zylinder detektiert zu werden.

Ein anderes Problem liegt vor, wenn der Quader eine quadratische Grundfläche hat. Die entstehenden Linienwolken und deren Punktmenge, siehe Abbildung 4.32, können den Bedingungen einer Ellipse nachkommen.

## 2. Bestimmung der geometrischen Größen

In diesem Abschnitt werden die Elemente  $e_i \in O$  bearbeitet, die als Quader oder Zylinder aus dem ersten Schritt *Bestimmung des Grundkörpers* identifiziert worden sind.

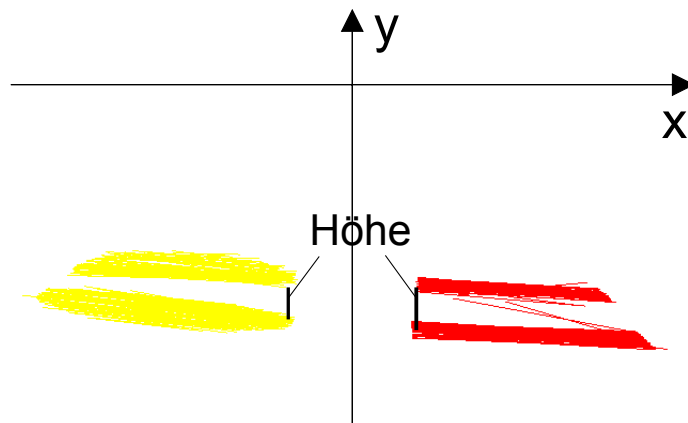


Abbildung 4.33: Bestimmung der Höhe mittels der Punkte aus der Objekt- und Verdeckungslinienwolke.

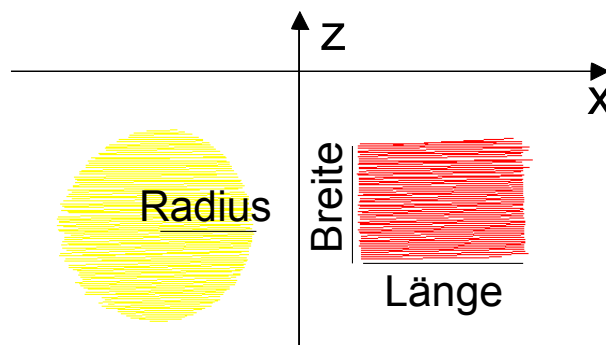


Abbildung 4.34: Der Radius wird durch die Länge einer mittleren Objektlinienwolke bestimmt. Die Breite und Länge des Quaders werden über die Objektlinienwolke festgelegt.

Ein Quader ist durch die drei Parameter, **Breite**, **Länge** und **Höhe**, festgelegt. Ein gerader Kreiszylinder ist durch zwei Parameter, **Höhe** und **Radius**, festgelegt.

Die Breite des Quaders wird durch die Differenz zwischen dem z-Wert der ersten Objektlinie und dem z-Wert der letzten Objektlinie berechnet. Diese Methode ist bei einem Scanverfahren, deren Linien in einem Abstand von über 2 cm liegen, nicht geeignet. Die Länge eines Quaders wird durch das arithmetische Mittel einiger Objektlinienlängen festgelegt (Abbildung 4.34). Die Quaderhöhe und die Zylinderhöhe werden durch die y-Differenz zwischen einem Punkt der Objektlinienwolke und einem Punkt der Verdeckungslinienwolke bestimmt, siehe Abbildungen 4.33. Der Radius wird durch die Länge der mittleren Objektlinie bestimmt (Abbildung 4.34).

### 3. Bestimmung der Lage im Raum

Für die Elemente  $e_i \in O$ , die als Quader oder Zylinder modelliert werden, muss in Java 3D ein Punkt im 3D Raum bestimmt werden, um das Objekt an gewünschter Stelle zu positionieren.

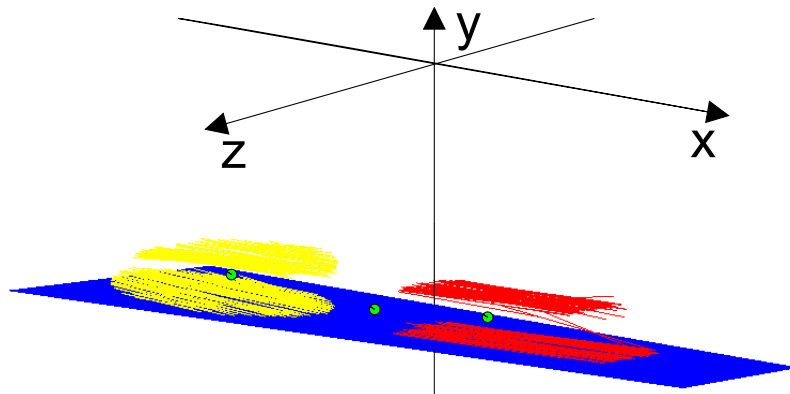


Abbildung 4.35: Die eingezeichneten grünen Punkte entsprechen den Mittelpunkten der Objekte bzw. der Tischfläche.

Für die Bestimmung dieses Punktes werden die Linienwolken des Elementes  $e_i$  verwendet und ein Punkt, der sich im mittleren Bereich der Linien befindet, bestimmt. In der Abbildung 4.35 sind die ermittelten Punkte von zwei Objekten und der Tischfläche grün eingezeichnet.

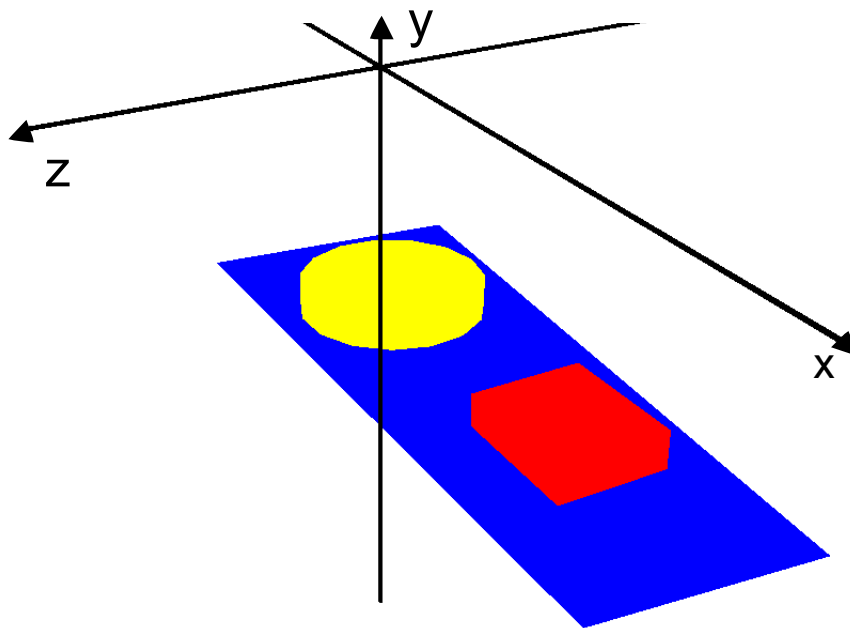


Abbildung 4.36: 3D Modell einer Tischezene.

### 4.3.3 Modellierung unbekannter Objekte

Wenn  $e_i \in O$  als unbekannt gilt, dann wird die Oberfläche für das Element  $e_i$  in Java 3D aus viereckigen Polygonen modelliert. Aus der Punktmenge der Objektlinienwolke  $CO_i$  werden die Eckpunkte der viereckigen Polygone erstellt. Zwischen allen aufeinanderfolgenden Linien der Objektlinienwolke wird ein viereckiges Polygon gezogen, siehe Abbildung 4.37(b). Das Ergebnis wird auch als Grundfläche genutzt, indem aber die y-Werte der Punkte durch die y-Werte der Punkte der Verdeckungslinienwolke  $CV_i$  ersetzt werden. Danach werden viereckige Polygone zwischen der Grundfläche und der Deckfläche gezogen, sodass eine geschlossene Objektoberfläche entsteht, wie in Abbildung 4.37 (d) und (e).



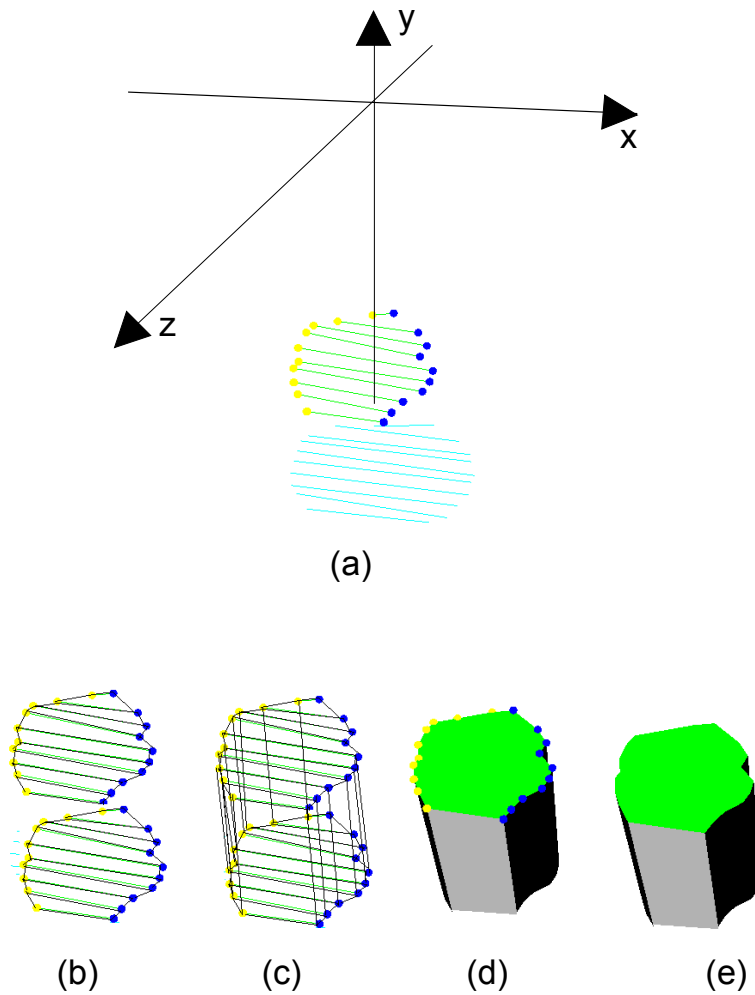


Abbildung 4.37: Oberflächenmodellierung eines unbekannten Objektes.

- (a) Objektlinienwolke (grün) mit Punktmenge, Verdeckungs­linienwolke (blau).  
 (b) Modellierung der Deck- und Grundfläche. (c) Modellierung der Seitenflächen.  
 (d) Modellierte Objekt­oberfläche und Punktmenge der Objekt­linienwolke. (e) Modellierte Objekt­oberfläche.



# Kapitel 5

## Implementierung des URG-PA10 3D Lasersystems

In diesem Kapitel werden einige Implementierungsaspekte des URG-PA10 3D Lasersystems, dessen konzeptionelle Entwurfsgrundlagen und viele Implementierungsdetails im vorherigen Kapitel beschrieben sind, dargestellt.

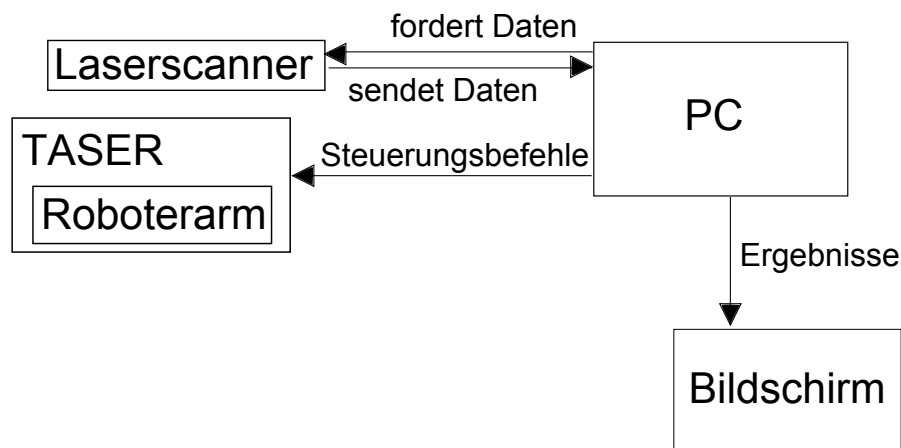


Abbildung 5.1: Hardwarekomponenten des Systems

Die Abbildung 5.1 zeigt, dass der Roboterarm und der Laserscanner über einen PC gesteuert werden. Der Roboterarm kommuniziert über eine ArcNet-Schnittstelle mit dem Pentium PC des TASERs. Der PC sendet Steuerungsbefehle an den PC des TASERs über eine drahtlose Verbindung,

sodass die Steuerungsbefehle über die ArcNet-Schnittstelle an den Roboterarm weitergeleitet werden. Der Laserscanner ist mit dem PC über einen USB-Anschluss verbunden. Die Datenaufforderung und das Senden der Messdaten erfolgen über die USB-Verbindung. Der PC empfängt und speichert die Daten. Die Daten werden analysiert und ausgewertet, um ein 3D Modell an einem Bildschirm auszugeben.

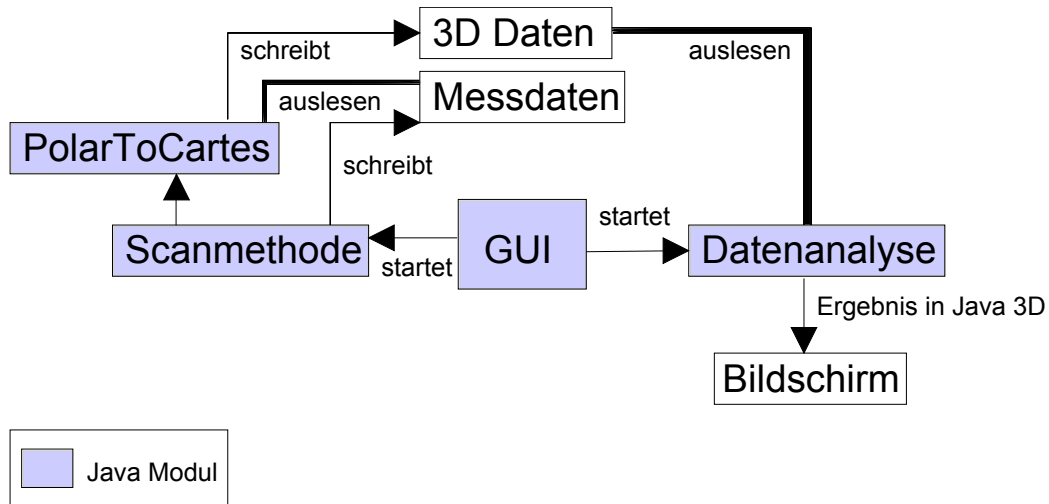


Abbildung 5.2: Softwarekomponenten



Abbildung 5.3: GUI zur Steuerung des URG-PA10 3D Lasersystems

Die Abbildung 5.2 zeigt die Softwaremodule, die für das URG-PA10 3D Lasersystem in Java implementiert sind. Die GUI (graphical user interface), die in Abbildung 5.3 dargestellt ist, ist eine graphische Benutzeroberfläche zur Steuerung des Systems. Aus der GUI können die implementierten Scanmethoden, Scan-Mit-Stop und Scan-Ohne-Stop, und die Datenanalyse gestartet werden. Bei der Datenanalyse ist es dem Anwender möglich, den

Scanbereich selbst festzulegen. Je nach Roboterlage und auch der Tischszene muss der Scanbereich angepasst werden. Das Starten einer Scanmethode löst einen automatisch laufenden Vermessungsprozess aus, indem der Roboterarm gesteuert wird und der Laserscanner aufgefordert wird, Daten zu senden. Die empfangenen Daten werden in einer Datei abgespeichert. Sobald die Scanmethode endet und alle Messdaten in der Datei liegen, werden die Messdaten automatisch von dem Java Modul **PolarToCartes** ausgelesen und in 3D Daten transformiert. Die 3D Daten werden zeilenweise in eine Datei abgelegt. Liegen die 3D Daten vor, so kann von der GUI aus die Datenanalyse gestartet werden. Die Datenanalyse ist ein automatisch ablaufender Prozess, der als erstes die 3D Daten aus der Datei ausliest. Auf die 3D Daten wird dann die dreistufige Clusteranalyse, die in Kapitel 4 beschrieben ist, angewandt. Die Clusteranalyse zerlegt die 3D Daten in Cluster. Jedes Cluster entspricht genau einem Objekt auf der Tischszene. Durch ein dreistufiges Wissenextraktionsverfahren werden geometrische Eigenschaften und die Lokalität für jedes Cluster bestimmt, um eine Oberfläche zu modellieren und das Objekt richtig in einen 3D Raum zu transformieren. Die Ergebnisse der Datenanalyse werden in Java 3D am Bildschirm ausgegeben.

## 5.1 Implementierung der Scanmethoden

### Scan-Mit-Stop

Die Steuerung des Armes erfolgt über eine Java-API, die von Dr. Martin Weser an der Universität Hamburg, Informatik im Arbeitsbereich TAMS erstellt worden ist. Die Steuerung des Hokuyo URG-LX04 ist in C,C++ geschrieben. Um Zugriff auf den Laserscanner für die Datenabfrage zu haben, wird eine Klasse in Java geschrieben, die eine JNI-Schnittstelle enthält. Mit dieser JNI-Schnittstelle ist es möglich, dass die in Java implementierten Scanmethoden den Laserscanner auffordern können, Messdaten zu senden.

Die Scan-Mit-Stop Methode ist durch einen Java-Thread realisiert. Dieser Thread enthält eine FOR-Schleife. In dieser FOR-Schleife wird der Arm schrittweise von Stützpunkt zu Stützpunkt aufgefordert sich zu bewegen und jeweils einige Sekunden zu warten, sodass der Laserscanner an jedem

Abstand in mm	Zeit in min:sek	Anzahl der Scans
1	20:52	461
2	14:54	231
4	6:12	116
5	4:54	93
10	2:12	47
20	1:10	24
23	1:03	21
46	0:36	11
92	0:22	6
115	0:19	5
230	0:14	3
460	0:11	2

Tabelle 5.1: Abstandswerte für die Scan-Mit-Stop Methode und die Messdauer.

Stützpunkt in einer ruhigen Lage einen Scan durchführen kann. In der FOR-Schleife sind der Empfang der Daten und auch das Schreiben der Daten in eine Datei implementiert. Die GUI bietet dem Anwender die Möglichkeit, selbst einen Abstandswert für die Scan-Mit-Stop Methode einzutragen. Die Auswahl der Eingabewerte für den Abstand ist auf die Teilermenge von 460 eingeschränkt.

$$T_{460} = \{1, 2, 4, 5, 10, 20, 23, 46, 92, 115, 230, 460\} \quad (5.1)$$

Die Tabelle 5.1 zeigt, je kleiner der Abstandswert für die Scanmethode ist, desto länger ist die Messdauer zum Vermessen der realen Tischszene.

### Scan-Ohne-Stop

Die Implementierung der Scan-Ohne-Stop Methode ist durch eine Java-Klasse, die zwei Threads hintereinander startet, realisiert. Der erste Thread fordert den Roboterarm auf, sich entlang der Bewegungsbahn in einer konstanten Geschwindigkeit zu bewegen. Der zweite Thread startet eine WHILE-Schleife.

---

Geschwindigkeitswert	Zeit ins sekunden	Scananzahl
0.01	235.4	592
0.05	48.3	122
0.1	24.9	63
0.15	17.0	43
0.2	13.1	34
0.25	10.8	28
0.3	9.2	24
0.35	8.0	21
0.4	7.2	19
0.45	6.6	17
0.5	6.1	16
0.55	5.6	15
0.6	5.3	14
0.65	5.0	13
0.7	4.7	12
0.75	4.5	12

Tabelle 5.2: Kleine Geschwindigkeitswerte der Scan-Ohne-Stop Methode führen zu einer höheren Anzahl von Scans.

Im Rumpf der WHILE-Schleife wird der Laserscanner aufgefordert, einen Scan durchzuführen und die Daten zu senden. Die Daten werden in einer Datei abgelegt. Der Abbruch der WHILE-Schleife erfolgt, sobald der Roboterarm am Ende der Bewegungsbahn angekommen ist. In der WHILE-Schleife ist daher eine Statusabfrage integriert, die den Roboterarm auffordert, seine genaue Positionslage anzugeben. Für die Scan-Ohne-Stop Methode kann ein Geschwindigkeitswert, der zwischen 0.01 und 0.75 liegt, eingetragen werden, um die Geschwindigkeit des Roboterarmes festzulegen.

Die Tabelle 5.2 zeigt das Verhältnis zwischen der Geschwindigkeit und der daraus resultierenden Scananzahl.

## 5.2 Datenverarbeitung

Die Algorithmen und Analyseschritte der Clusteranalyse auf Seite 53 und die Verarbeitungsschritte der Wissensextraktion auf Seite 60 sind wie beschrieben in Java implementiert worden und erzeugen ein 3D Modell der Tischszene. Dieses 3D Modell wird in einem Frame von Java 3D angezeigt. In diesem Frame ist ein Panel eingebaut und ermöglicht dem Anwender einige Ergebnisse einzelner Analyseschritten zu betrachten, wie z.B. die Linienwolken auf Seite 59.

Das Frame und das integrierte Panel sind in Abbildung 5.4 zu sehen.

In der Phase der Datenauswertung und Implementierung der Clusteranalyse sind konstant bleibende Schwellwerte bestimmt worden, die zur Segmentierung und Gruppierung der 3D Punktwolke verwendet werden. Einige der Schwellwerte hätten für Punktwolken mit einer spärlichen Punktedichteverteilung anders gesetzt werden müssen, da sie weniger Informationen enthalten. Die Anpassung der Schwellwerte für Punktwolken mit niedriger Punktedichte ist nicht gemacht worden, da zu wenig Informationen zur Unterscheidung von Objekten, ob Zylinder oder Quader, vorliegen.



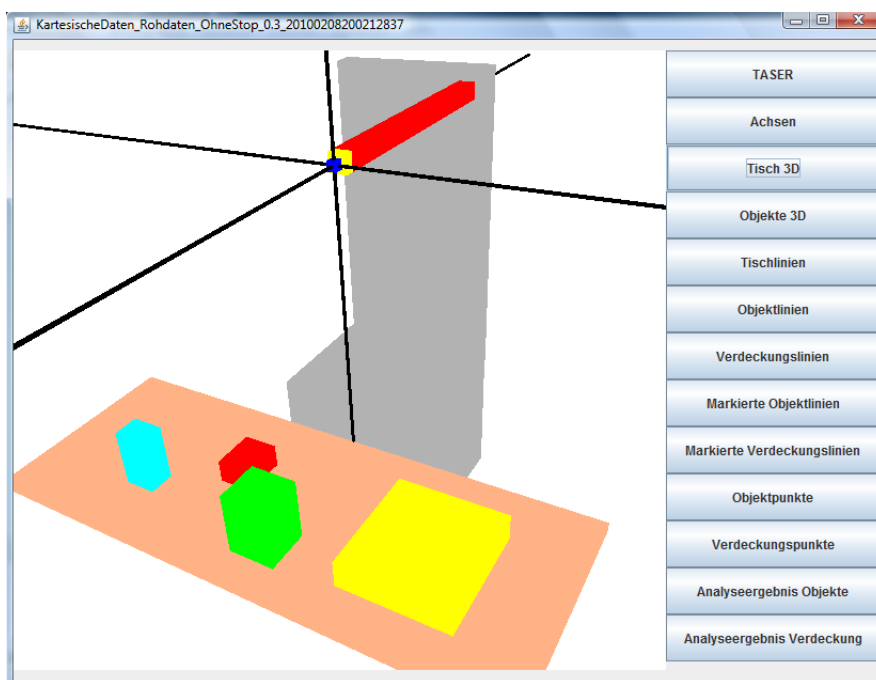
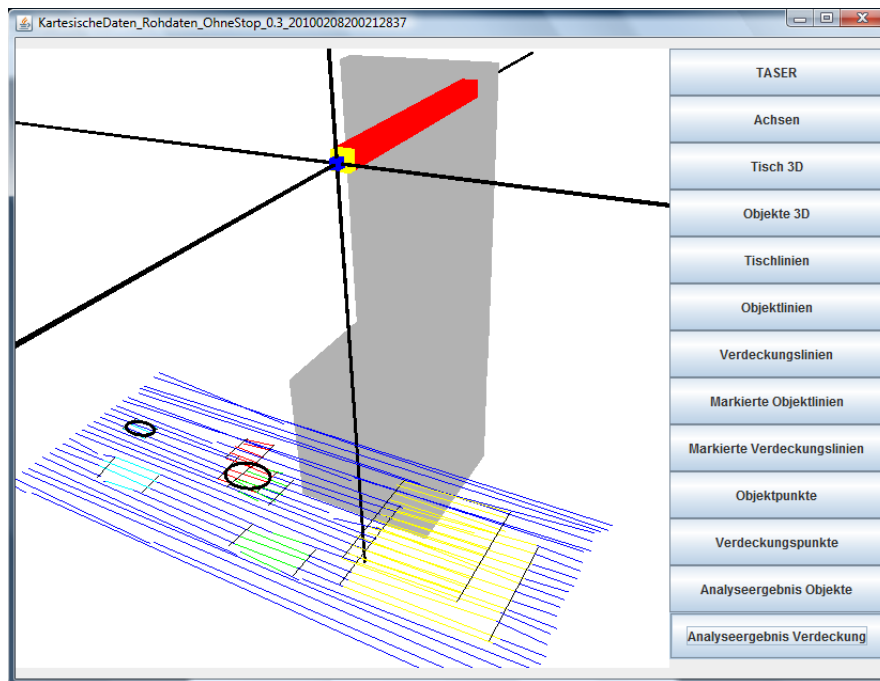


Abbildung 5.4: Java 3D. Rekonstruktion und Modellierung einer real vermessenen Tischszene



# Kapitel 6

## Experimentelle Ergebnisse und Auswertung

Ein gutes 3D Rekonstruktionsergebnis, das das URG-PA10 3D Lasersystem liefern kann, ist gegeben, wenn folgende Punkte erfüllt werden.

1. Alle Objekte der Tischszene sind detektiert und als 3D Modell modelliert.
2. Jedem detektierten Objekt wird die richtige Grundform zugewiesen. Alle realen quaderförmigen Objekte der Tischszene werden vom System auch als Quader erkannt und alle realen zylinderförmigen Objekte der Tischszene werden vom System auch als Zylinder erkannt.
3. Die berechneten geometrischen Größen für jedes 3D Objekt sind den realen Größen der Objekte gleich bzw. annähernd gleich.
4. Jedem Objekt ist eine genaue Lage im Raum zugeordnet, sodass die Objekte im 3D Raum genau wie in der realen Tischszene an der richtigen Stelle positioniert werden.

Im Folgenden werden Ergebnisse der Scan-Mit-Stop Methode und der Scan-Ohne-Stop Methode gezeigt. Die Ergebnisse beziehen sich auf die reale Tischszene, die in den Abbildungen 6.1 und 6.2 dargestellt ist. Die Tischszene hat zwei Quader-Objekte und zwei Zylinder-Objekte. Die Ausgangssituation für alle Scanmessvorgänge ist die gleiche Szene. Es werden keine Veränderungen an

---

der Tischszene vorgenommen, und die Position des Roboters bleibt unverändert, sodass alle Messungen aus einer gleichen Lage erfolgen und der Scanbereich für alle Messergebnisse gleich ist.

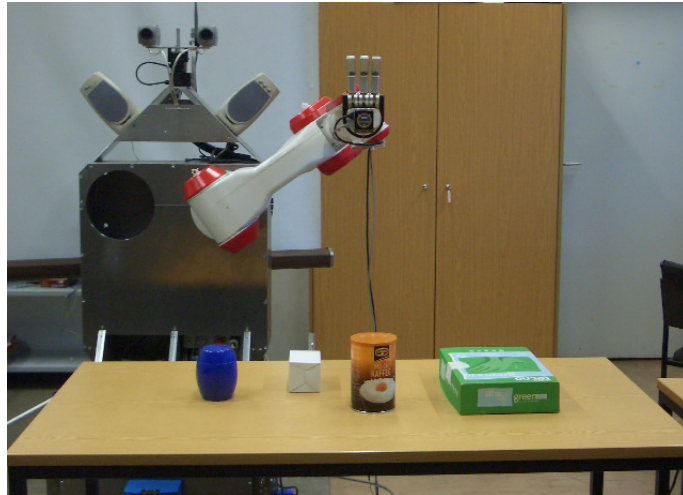


Abbildung 6.1: Reale Tischszene.



Abbildung 6.2: Reale Tischszene.

## 6.1 Experimente und Ergebnisse der Scan-Mit-Stop Methode

In diesem Abschnitt werden Untersuchungen der Scan-Mit-Stop Methode vorgenommen. Eine Scan-Mit-Stop Methode wird mit gleichem Abstandswert dreimal hintereinander ausgeführt, um die Ergebnisse einer Vermessungsmethode unter gleichen Konditionen zu vergleichen.

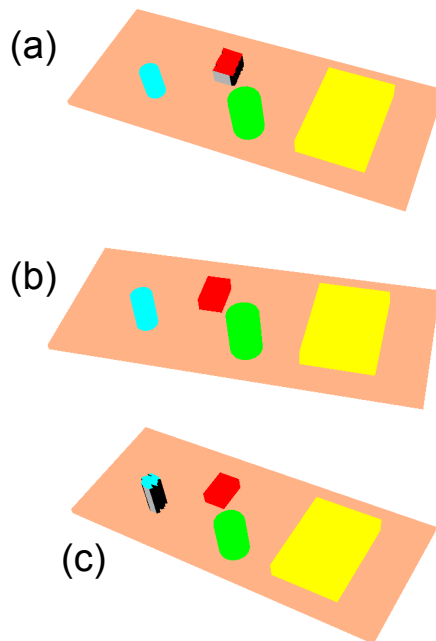


Abbildung 6.3: Ergebnisse der Scan-Mit-Stop Methode mit dem Abstandswert von 0.5 cm.

Abstand in cm (Anzahl der Scans)	0.5 (93)	1 (46)	4.6 (11)
Ergebnis 1	1 Q, 2 Z, 1 U	2 Q, 2 U	3 Q
Ergebnis 2	2 Q, 2 Z	2 Q, 1 Z, 1 U	3 Q
Ergebnis 3	2 Q, 1 Z, 1 U	2 Q, 2 U	3 Q

Tabelle 6.1: Ergebnisse der Scan-Mit-Stop Methode. Abkürzungen: Quader (= Q), Zylinder (= Z), unbekannt (= U)

Tabelle 6.1 zeigt die drei Ergebnisse der Scan-Mit-Stop Methode mit gleichem Abstandswert. Die Ergebnisse beziehen sich auf die detektierten Objekte und

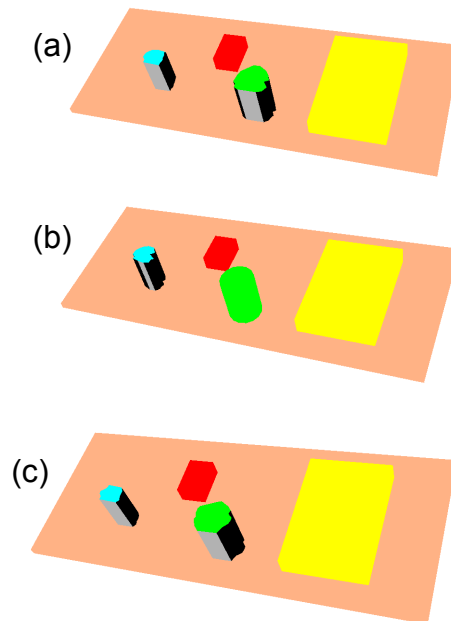


Abbildung 6.4: Ergebnisse der Scan-Mit-Stop Methode mit dem Abstandswert von 1 cm.

deren Grundform. Jedem detektierten Objekt wird entweder der Grundkörper Quader oder Zylinder zugewiesen. Es kommt aber auch vor, dass ein Objekt weder einem Quader noch Zylinder zugewiesen wird. Dieses Objekt wird dann als unbekannt deklariert und als Polyeder bestehend aus viereckigen Polygonen modelliert, siehe Seite 72.

In den Abbildung 6.3, 6.4 und 6.5 sind die Ergebnisse der einzelnen Messreihen zu sehen.

Es ist ersichtlich, dass die Scanmethode mit dem kleinsten Abstandswert, also 0.5 cm, die besten Ergebnisse liefert, da die Tischszene insbesondere der zweiten Messung, siehe 6.3 (b) gut approximiert rekonstruiert worden ist. Die zwei Quader-Objekte und die zwei Zylinder-Objekte sind richtig erkannt und dargestellt. Aus allen Messergebnissen der Scanmethode mit dem Abstandswert von 4.6 cm geht hervor, dass drei Objekte von vier Objekten detektiert und als Quader-Objekte modelliert werden.

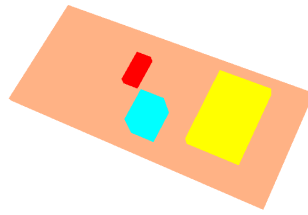


Abbildung 6.5: Die Scan-Mit-Stop Methode mit dem Abstandswert von 4.6 cm liefert bei allen drei Messreihen das gleiche Ergebnis.

## 6.2 Experimente und Ergebnisse der Scan-Ohne-Stop Methode

Die Scan-Ohne-Stop Methode ist ein schnelles Messverfahren, das eine schnelle Durchführung von mehreren Messreihen erlaubt. Im Folgenden werden einige Ergebnisse der Scan-Ohne-Stop Methode mit unterschiedlichen Armgeschwindigkeiten vorgelegt. Jede Scanmethode mit einer festgesetzten Geschwindigkeit ist zehnmal hintereinander auf die gleiche Tischszene aus Abbildung 6.1 ausgeführt worden.

In den Abbildung 6.6, 6.7, 6.8 und 6.9 sind die Ergebnisse der einzelnen Messreihen zu sehen.

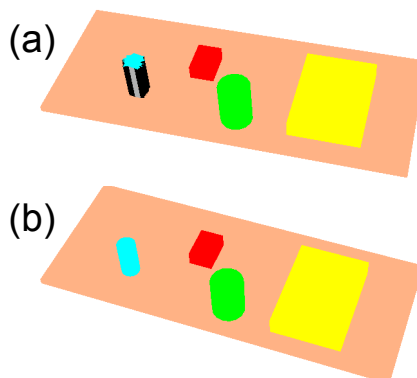


Abbildung 6.6: Die Scan-Ohne-Stop Methode mit dem Geschwindigkeitswert von 0.05 detektiert bei allen zehn durchgeführten Messungen alle vier Objekte. Die zwei Quader werden in den meisten Messergebnissen korrekt erkannt, während bei den Zylinder Objekten eins von beiden meist unerkannt bleibt, wie in (a).

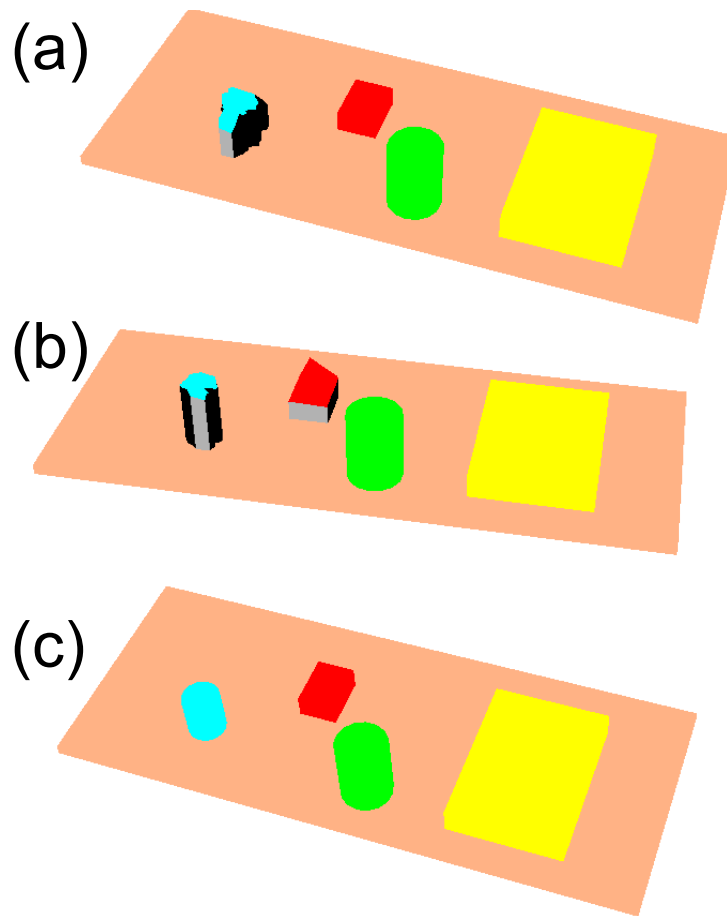


Abbildung 6.7: Die Scan-Ohne-Stop Methode mit dem Geschwindigkeitswert von 0.1 detektiert bei allen zehn durchgeführten Messungen alle vier Objekte. Die zwei Quader werden in den meisten Messergebnissen korrekt erkannt, während bei den Zylinder Objekten eins von beiden meist unerkant bleibt, wie in (a) und (b).

Gute Ergebnisse der Scan-Ohne-Stop Methode sind mit dem Geschwindigkeitswert 0.1 erzielt worden. Die Scanmethode liefert 63 Scans. Die Scans liegen in einem Abstand von ca. 7.4 mm entlang der Bewegungsbahn voneinander entfernt.



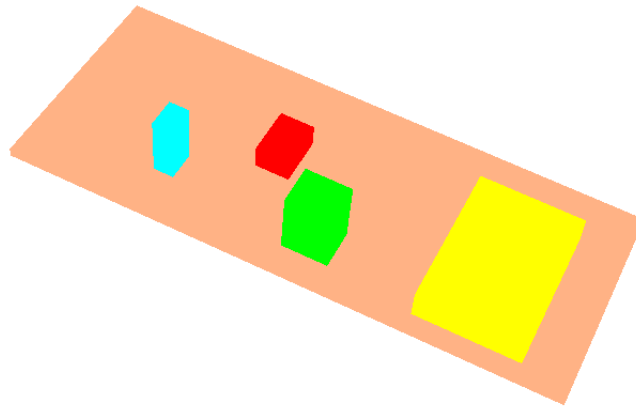


Abbildung 6.8: Die Scan-Ohne-Stop Methode mit dem Geschwindigkeitswert von 0.3 detektiert bei allen zehn durchgeführten Messungen alle vier Objekte. Alle Objekte werden als Quader identifiziert.

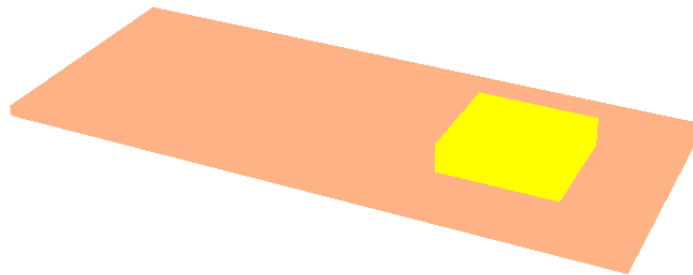


Abbildung 6.9: Die Scan-Ohne-Stop Methode mit dem Geschwindigkeitswert von 0.6 detektiert bei allen zehn durchgeführten Messungen nur ein Objekt von den vier Objekten. Das detektierte Objekt ist das größte Objekt.

### 6.3 Vergleich der beiden Scanmethoden

Beide Scanmethoden liefern unter bestimmten Konditionen, wie z.B. einem bestimmten Abstandswert bzw. Geschwindigkeitswert, gute Ergebnisse. Liegt die Bewegungsbahn ca. 60 cm über der Tischfläche, liefert die Scan-Ohne-Stop Methode mit einer Armgeschwindigkeit von 0.1 und die Scan-Mit-Stop Methode bei einem Abstand von 5 mm gute Ergebnisse. Während der Messvorgang der Scan-Ohne-Stop Methode bei einer Armgeschwindigkeit von 0.1 insgesamt 63 Scans innerhalb von 25 Sekunden dauert, benötigt die Scan-Mit-Stop Methode bei einem Abstand von 5 mm das Zwölfwache an Zeit, ca. 4:54 min, und liefert

93 Scans. Vergleicht man die beiden Scanmethoden, so ist klar, dass die Scan-Ohne-Stop Methode zu bevorzugen ist, da die Messdauer sehr kurz ist und trotz weniger Messdaten gleich gute Ergebnisse erzielt werden. Gerade sensorbasierte Systeme sollten in der Lage sein, schnell Informationen über eine Umgebung zu ermitteln, um die Daten auszuwerten. Denn reale Umgebungen können sich durch andere Umstände oder auch Störfaktoren verändern, die vom Sensorsystem sofort erkannt werden sollten, damit entsprechend reagiert werden kann.

Anhand der experimentellen Ergebnisse hat sich herausgestellt, dass beide Scanvarianten für die Detektion und Modellierung von quaderförmigen Objekten gut geeignet sind.

### Weitere Ergebnisse

In diesem Abschnitt werden einige Ergebnisse von anderen Tischszenen, die mit der Scan-Ohne-Stop Methode (Geschwindigkeit 0.1) vermessen worden sind, gezeigt.

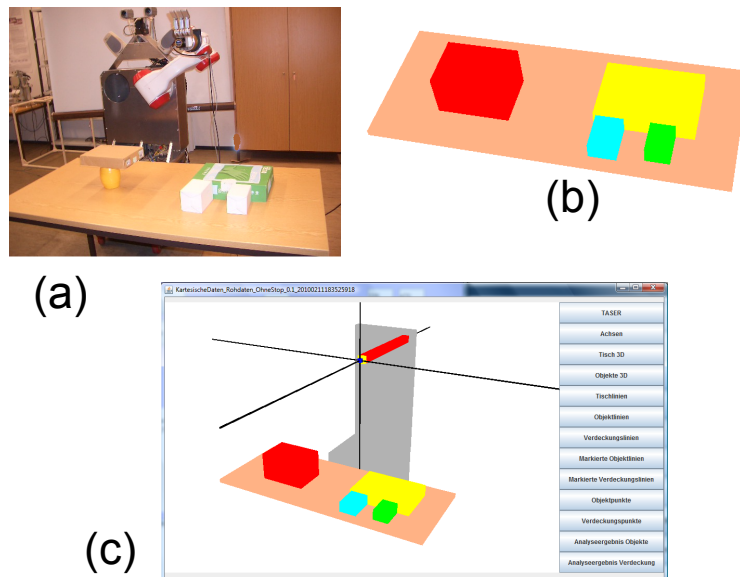


Abbildung 6.10: (a) Reale Tischszene. (b) 3D Rekonstruktion. (c) 3D Modell im Java 3D Frame.

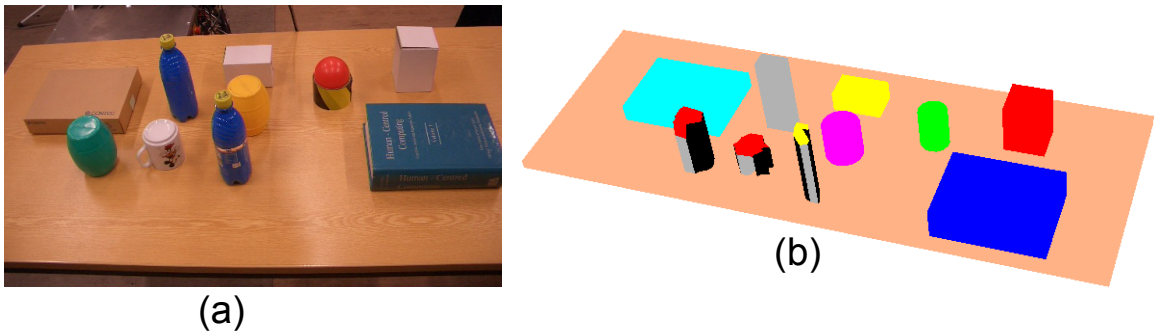


Abbildung 6.11: (a) Reale Tischszene. (b) 3D Rekonstruktion.

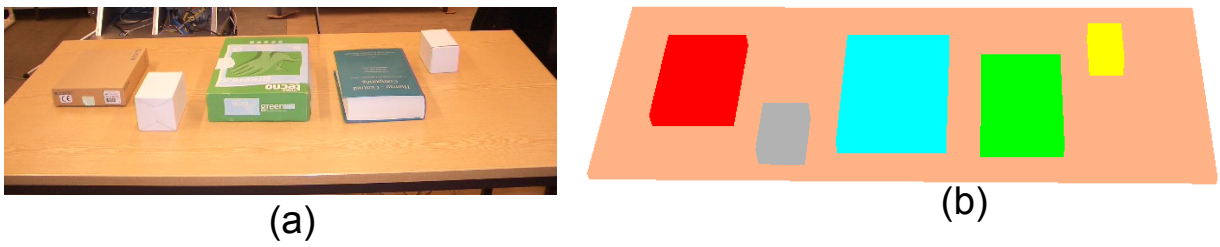


Abbildung 6.12: (a) Reale Tischszene. (b) 3D Rekonstruktion.

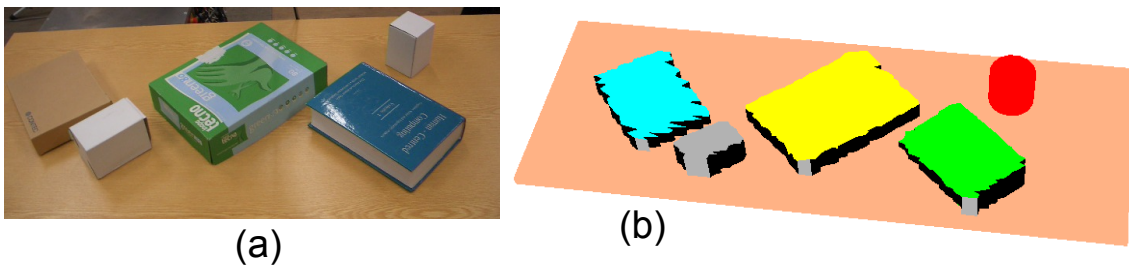


Abbildung 6.13: (a) Reale Tischszene. (b) 3D Rekonstruktion.



# Kapitel 7

## Fazit und Ausblick

In dieser Arbeit wurde ein 3D Lasersystem entwickelt, das Quader- und Zylinder-Objekte, die sich auf einer Tischszene befinden, detektieren und sie als Quader bzw. Zylinder modellieren kann. Weiterhin kann das System die genaue Lage der einzelnen Objekte im Raum bestimmen und die Tischszene als 3D Modell an einem Bildschirm ausgeben. Schlussfolgernd kann man sagen, dass ein lineares Scanverfahren zur Detektion von quader- und zylinderförmigen Objekten geeignet ist.

Dieses System kann erweitert und optimiert werden und als Grundlage für andere Robotersysteme, wie z.B. einem Greifsystem, Überwachungssystem, etc. verwendet und angepasst werden.

Möglichkeiten für die Erweiterungen des URG-PA10 3D Lasersystems sind:

- Entwicklung anderer Scanmethoden, um komplexe Objekte detektieren und modellieren zu können.
- Erweiterung der Clusteranalyse und den Wissensextraktionsalgorithmen, um andere primitive Körper, wie z.B. Pyramide, Kugel, Torus, etc., zu erkennen.
- Andere Verarbeitungsalgorithmen zur Analyse der 3D Punktwolke.

---

Das URG-PA10 3D Lasersystem kann als Grundlage für folgende Systeme dienen:

- Greifsystem. Die Tischszene kann in das Roboterkoordinatensystem des TASERs überführt werden, sodass dem Roboter das Wissen einer Tischszene samt den Objekten vorliegt. In dem Wissen sind die geometrischen Größen und die genaue Lage der einzelnen Objekte bekannt. Anhand dieser Daten kann eine Greifaktion mit dem Roboterarm PA10-6C und der Dreifingerhand BarrettHand unter der Berücksichtigung von Kollisionsgefahren berechnet werden.

- Überwachungssystem.

Beispiel für ein Überwachungssystem

In einem Museum soll eine Tischszene, auf der antike Gegenstände liegen, einmal die Stunde überwacht werden. Die Tischszene wird vom URG-PA10 3D Lasersystem vermessen und ausgewertet. Die Messergebnisse und Auswertungen werden gespeichert und mit den weiteren Messungen, die einmal die Stunde stattfinden, verglichen. Sobald eine Änderung in der Anzahl der Objekte oder gewissen geometrischen Größen vorkommt, kann das System ein Alarmsignal auslösen.

# Literaturverzeichnis

- [1] O. WULF AND B. WAGNER: *Fast 3D scanning methods for laser measurement systems*. International Conference on Control Systems and Computer Science, Bucharest, Romania, 2003
- [2] H. SURMANN, A. NÜCHTER, K. LINGEMANN, AND J. HERTZBERG: *6D SLAM A Preliminary Report on Closing the Loop in Six Dimensions*. Proc. of the 5th IFAC Symp. on Intelligent Autonomous Vehicles, (IAV), Lisabon, Portugal, 2004
- [3] T. UEDA, H. KAWATA, T. TOMIZAWA, A. OHYA, S. YUTA: *Mobile SOKUIKI Sensor System: Accurate RAnge Data Mapping System with Sensor Motion*. 3rd International Conference on Autonomous Robots and Agents (ICARA), 2006
- [4] V.T. NGUYEN, A. MARTINELLI, N. TOMATIS, R. SIEGWART. : *A comparison of line extraction algorithms using 2D laser rangefinder for indoor mobile robotics*. IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, Edmonton, Canada, 2005
- [5] C. PREMEBIDA AND U. NUNES: *Segmentation and geometric primitives extraction from 2D laser range data for mobile robot applications*. Robotica 2005, Institute of Systems and Robotics ISR University of Coimbra, Coimbra, Portugal
- [6] G. VOSSELMAN, B. GORTE, G. SITHOLE, UND T. RABBANI: *Recognising structure in laser scanner point clouds*. International Conference on Laser-Scanners for Forest and Landscape Assessment and Instruments, Processing Methods and Applications, Freiburg, Germany, 2004

- [7] J. ALFERES, E. ZALAMA , J. GÓMEZ-GARCÍA-BERMEJO:  
*Reconstruction of Three Dimensional Models of Environments with a Mobile Robot*. Proceedings of the 2005 IEEE, International Conference on Robotics and Automation Barcelona, Spain, April 2005
- [8] B. GORTE:  
*Segmenation of TIN-structured surface models*. Symposium on Geospatial Theory, Processing and Applications; Working Group IV/6, Ottawa, Canada, 2002
- [9] J. EICHLER UND H.-J. EICHLER  
*Laser: Bauformen, Strahlführung, Anwendungen* Springer, 5. aktualisierte Ausgabe, 2003
- [10] <http://www.qm-infocenter.de/>  
Messprinzip Triangulation (2009)
- [11] <http://www.intergeo.de/archiv/2007/Riegl.pdf>  
Riegl - Laserscanning (2009)
- [12] J. PAULI:  
*Autonome Robotersysteme*. Skript zur Vorlesung, Lehrstuhl für Kognitive Systeme, Christian-Albrechts-Universität zu Kiel, Institut für Informatik und Praktische Mathematik, Letzte Änderung 24. Mai 2006
- [13] J. CRAIG  
*Introduction to Robotics - Mechanics and Control* 3. Auflage, Prentice Hall, 2003.
- [14] <http://de.wikipedia.org/wiki/Polarkoordinaten>  
Bild Zylinderkoordinatensystem (2009)
- [15] F. KLAWONN  
*Grundkurs Computergrafik mit Java - Die Grundlagen verstehen und einfach umsetzen mit Java 3D*. Friedr. Vieweg Sohn Verlag /GWV Fachverlage GmbH, Wiesbaden 2005
- [16] L. RADE, B. WESTERGRENN:



- Springers Mathematische Formeln - Taschenbuch für Ingenieure, Naturwissenschaftler, Wirtschaftswissenschaftler.* Springer-Verlag Berlin Heidelberg, 2. Auflage, 1997
- [17] H.-J. BUNGARTZ, M. GRIEBEL, C. ZENGER  
*Einführung in die Computergraphik: Grundlagen, geometrische Modellierung, Algorithmen 2.* überarbeitete und erweiterte Auflage; Braunschweig, Vieweg, 2002
- [18] K. BRUNS, K. MEYER-WEGENER  
*Taschenbuch der Medieninformatik* Fachbuchverlag Leipzig im Carl Hanser Verlag München Wien, 2005
- [19] I. BIEDERMAN *Recognition by components: A Theory of Human Image Understanding.* State University of New York at Buffalo, American Psychological Association, Inc., 1987
- [20] W. WILKE:  
*Segmentierung und Approximation großer Punktwolken.* Dissertation, Fachbereich Mathematik, Technische Universität Darmstadt, 2000
- [21] U. FLICK  
*Triangulation, eine Einführung* 1. Auflage, Wiesbaden, Verl. für Sozialwiss., 2004
- [22] F. WELLER:  
*Geometrische Algorithmen in der Flächenrückführung.* Dissertation, FB 04, Universität Dortmund, 2000
- [23] H. STAHL: *Clusteranalyse großer Objektmengen mit problemorientierten Distanzmaßen.* Verlag Harri Deutsch Thun, Frankfurt am Main, 1985
- [24] F. PUPPE: *Wissensmanagementsysteme.* Skript zur Vorlesung, Informatik, Universität Würzburg, Wintersemester 2002/2003
- [25] M. ESTER, J. SANDER: *Wissensmanagementsysteme.* Knowledge Discovery in Databases. Springer, 2000
- [26] H. SPÄTH: *Cluster-Formation und -Analyse, Theorie, FORTRAN-Programme, Beispiele.* Oldenbourg Verlag GmbH, München, 1983

- [27] B. EVERITT *Cluster analysis* London, Arnold, 1993
- [28] T. MARECH: *Mathematik-Verknüpfung von 2D- und 3D-Punktwolken*. Dissertation, Gießen, Justus-Liebig-Universität, FB 07 Mathematik und Informatik, Physik, Geographie, 2006
- [29] J. ENCARNAÇÃO, W. SRASSER, R. KLEIN *Graphische Datenverarbeitung 1* R. Oldenbourg Verlag, 5. Auflage, 1996
- [30] J. ENCARNAÇÃO, W. SRASSER, R. KLEIN *Graphische Datenverarbeitung 2* R. Oldenbourg Verlag, 5. Auflage, 1997
- [31] *Java 3D Tutorial* <http://ilias.fh-stralsund.de/letzter> Aufruf 01.12.2009
- [32] T. BAIER *Lernen der Handhabung von Alltagsgegenständen im Kontext eines Service-Roboters* Dissertation, Universität Hamburg, Informatik - Department Informatik, TAMS, Januar 2008
- [33] *TAMS Research Robotics*. TAMS, Informatik, Universität Hamburg <http://tams-www.informatik.uni-hamburg.de> letzter Aufruf 01.02.2010
- [34] J.E. LLOYD, V. HAYWARD: *Multi-RCCL Users Guide*.. Montreal, Quebec, Canada, 1992 <http://people.cs.ubc.ca/~lloyd/rccl.html> letzter Aufruf 01.02.2010
- [35] MORI, MAEJIMA, SANTOSH, MAEDA: *Scanning Laser Range Finder URG-04LX Specifications*. HOKUYO AUTOMATIC Co., Ltd., 2005
- [36] [http://www.hokuyo-aut.jp/02sensor/07scanner/urg\\_04lx.html](http://www.hokuyo-aut.jp/02sensor/07scanner/urg_04lx.html)  
<http://www.hokuyo-aut.jp/> - HOKUYO URG-LX04 (2009)
- [37] D. KLIMENTJEW, M. ARLI, J. ZHANG *3D Scene Reconstruction Based on a Moving 2D Laser Range Finder for Service-Robots*. In Proceedings of the IEEE International Conference on Robotics and Biomimetics (ROBIO), Guilin, Guangxi, China, December 18-22, 2009 (In press).

# Erklärung

Ich versichere, dass ich die vorstehende Arbeit selbstständig und ohne fremde Hilfe angefertigt und mich anderer als der im beigefügten Verzeichnis angegebenen Hilfsmittel nicht bedient habe. Alle Stellen, die wörtlich oder sinngemäß aus Veröffentlichungen entnommen wurden, sind als solche kenntlich gemacht.

Hamburg, den 18.02.2010