



Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Bachelorarbeit

Matthias Wegscheider

Charakterisierung und Realisierung einer
mikrokontrollerbasierten 3D-Erfassung,
basierend auf einem Laserscanner und
einer externen Schwenkeinheit

Matthias Wegscheider
Charakterisierung und Realisierung einer
mikrokontrollerbasierten 3D-Erfassung,
basierend auf einem Laserscanner und einer
externen Schwenkeinheit

Bachelorarbeit eingereicht im Rahmen der Bachelorprüfung
im Studiengang Technische Informatik
am Department Informatik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer : Prof. Dr. -Ing. Franz Korf
Zweitgutachter : Prof. Dr. rer. nat. Reinhard Baran

Abgegeben am 25. Juni 2008

Matthias Wegscheider

Thema der Bachelorarbeit

Charakterisierung und Realisierung einer mikrokontrollerbasierten 3D-Erfassung, basierend auf einem Laserscanner und einer externen Schwenkeinheit

Stichworte

Koordinatensysteme, Koordinatentransformationen, Anwenderschnittstelle, Lasermesssysteme, 2D-Laserscanner, Motorschwenkeinheit, Synchronisation, State-Machine, 3D-Visualisierung.

Kurzzusammenfassung

Gegenstand dieser Bachelorarbeit ist die Entwicklung einer 3D-Erfassung einer Messumgebung. Die 3D-Erfassung wird realisiert, indem zwei Einheiten, ein 2D-Laserscanner und eine Motorschwenkeinheit zu einer logischen 3D-Einheit bzw. zu einem logischen 3D-Laserscanner zusammengefasst werden. Für diese neue logische 3D-Einheit werden in dieser Arbeit die Anforderungen analysiert und weitestgehend in einem Prototypen umgesetzt. Zentraler Punkt bei der Erstellung des Prototyps war die Charakterisierung und die Entwicklung einer modular aufgebauten und leicht erweiterbaren 3D-Erfassungssteuerung, die auf einem von zwei Mikrokontrollern des Laserscanners ausgeführt wird.

Matthias Wegscheider

Title of the paper

Characterization and realization of a microcontroller based 3D-measurement, based on a laser scanner and an external swivel unit.

Keywords

Co-ordinate-system, co-ordinate transformation, user interface, laser measurement systems, 2D-laser scanner, motorized swivel unit, transformation, synchronization, state-machine, 3D-visualisation.

Abstract

This bachelor thesis includes the development of a 3D-measurement of a measurement environment. The 3D-measurement will be realized by a logic 3D-unit, which consists of a laser measurement system and a motorized swivel unit. The requirements of the new logic 3D-unit will be analyzed and a prototype will be created. The main focus on creating the prototype was the development and characterization of a 3D-measurement controller. The 3D-measurement controller has a modular system architecture, is easy expandable and will be executed on the second microcontroller of the laser measurement system.

Inhaltsverzeichnis

Tabellenverzeichnis	8
Abbildungsverzeichnis	9
1. Einführung	11
1.1. Motivation	11
1.2. Einsatzgebiete und verwandte Lösungen	14
1.3. Zielsetzung und Abgrenzung	15
1.4. Gliederung der Arbeit	16
2. Grundlagen	17
2.1. Kurzüberblick und Einordnung von 3D-Messmethoden	17
2.1.1. Passive Messmethoden	17
2.1.2. Aktive Messmethoden	18
2.2. Messen mit einem Laserscanner	19
2.2.1. Grundbegriffe	19
2.2.2. Das Prinzip der 3D-Erfassung mit einem Laserscanner	21
2.2.3. Einflüsse auf die Entfernungsmessung beim Laserscannen	23
2.3. Grundlagen der 3D-Koordinatenverarbeitung	25
2.3.1. Polar vs. Kartesisch	25
2.3.2. Lokales vs. weltliches Koordinatensystem	26
2.3.3. Koordinatentransformationen	27
2.4. Zusammenfassung	29
3. Rahmenbedingungen	30
3.1. Features der Akteure der 3D-Einheit	30
3.1.1. 2D-Laserscanner LRS-2100	30
3.1.2. Powercubeeinheit	31
3.2. Die Bewegung der 3D-Erfassung	31
3.3. Definitionen für die 3D-Erfassung	32
3.4. Interner Aufbau der 3D-Einheit	33
3.4.1. Entwicklung einer 3D-Erfassungssteuerung auf dem C32	33
3.4.2. Vorgaben für die Entwicklung auf dem C32	35

3.5. Anforderungen	38
3.5.1. Funktionale Anforderungen	38
3.5.2. Nicht funktionale Anforderungen	39
3.6. Randbedingungen für die Entwicklung	40
3.7. Zusammenfassung	40
4. Design und Realisierung	41
4.1. Übersicht über die 3D-Erfassung	41
4.1.1. Modularer Aufbau	41
4.1.2. Der Ablauf und die Module der 3D-Erfassung	41
4.1.3. Design und Realisierungsreihenfolge der Module	44
4.2. 3D-Anwenderschnittstelle	45
4.2.1. Der Aufbau des 3D-Service-Protokolls	45
4.2.2. Der Aufbau eines 3D-Datenprofils	46
4.2.3. 3D-Kommandos	48
4.3. State-Machine	50
4.3.1. Wahl des Harelautomaten	51
4.3.2. Ausdrücke des Harelautomaten und die Verwendung in der State-Machine	51
4.3.3. Zustände des Automaten der 3D-Erfassungssteuerung	52
4.4. Ansteuerung der Powercubeinheit	56
4.5. Step-Modus	57
4.5.1. Grundprinzip	57
4.5.2. Konzept und Realisierung des Step-Modus	57
4.6. Interpolations-Modus	60
4.6.1. Der 3D-Single-Scan	60
4.6.2. Erfassung einer Messumgebung	66
4.6.3. Offene Punkte für die Implementierung	69
4.7. 3D-Transformation	70
4.7.1. Definierung des lokalen Koordinatensystems der 3D-Einheit	70
4.7.2. Transformation in das lokale Koordinatensystem der 3D-Einheit	71
4.8. Zusammenfassung	74
5. Qualitätssicherung	75
5.1. Aufbau der Testanwendung	75
5.1.1. Testmodul Anwendersteuerung	75
5.1.2. Testmodule für einen 3D-Laserscan	75
5.2. Beschreibung der durchgeführten Tests	77
5.2.1. Test der Anwenderschnittstelle und der State-Machine	77
5.2.2. Test der 3D-Erfassung im Step-Modus	78
5.3. Zusammenfassung der Testergebnisse	80

5.4. Zusammenfassung	84
6. Zusammenfassung und Ausblick	85
6.1. Bewertung der realisierten 3D-Erfassung	85
6.1.1. Erreichte Ziele	85
6.1.2. Offene Punkte und Verbesserungen	86
6.2. Erweiterungsmöglichkeiten	87
Literaturverzeichnis	88
A. Memory Mapping einer C32-Anwendung im Hello-World-Zustand	92
B. Errorcode-Tabelle der 3D-Erfassungssteuerung	93
C. Berechnung der Positionsabweichung der Powercubdreahachse im Step-Modus	94
D. 3D-Kommandos der Anwenderschnittstelle	95
E. Inhalt der CD	98

Tabellenverzeichnis

2.1. Remissionswerte in Abhängigkeit von der Art der Oberfläche [Kap. 3 Sick Ibeo GmbH, 2004a].	24
2.2. Übersicht über die Rotationen um die verschiedenen Achsen.	29
3.1. Der Speicherverbrauch bei einem 3D-Laserscan.	37
3.2. Literaturhinweise für die Komponenten der 3D-Einheit.	37
4.1. Darstellung eines 3D-Messpunktes unter den verschiedenen Optionen.	48
4.2. Die Daten, die für eine lineare Interpolation nötig sind.	61
A.1. Memory-Mapping einer Hello-World-C32-Anwendung mit Spi-Treiber und Interface-Funktionen.	92
B.1. Errorcodetabelle der 3D-Einheit	93

Abbildungsverzeichnis

1.1. Vergleich des 3D-Laserscanners der Firma Lase und dieser Bachelorarbeit.	12
1.2. Abstraktes Schichtmodell der Verarbeitung von 3D-Daten.	15
2.1. Überblick über die aktiven Messmethoden [vgl. Kap.2 Gühring, 2002].	18
2.2. Links: Zeichnung des ersten Halbbildes. Mitte: Zeichnung des zweiten Halbbildes. Rechts: Fertig gezeichnetes Bild [Wikipedia.org, 2008b].	20
2.3. 2D-Messung einer Punktlinie.	22
2.4. 3D-Messung, bestehend aus mehreren 2D-Punktlinien.	22
2.5. Diffuse Reflektion einer rauen Oberfläche [Kap.3 Sick Ibeo GmbH, 2004a].	24
2.6. Links: schräges Laserbestrahlen \Rightarrow schlechter Reflektionswinkel Rechts: senkrecht Laserbestrahlen \Rightarrow guter Reflektionswinkel [Kap.3 Sick Ibeo GmbH, 2004a].	24
2.7. Das katesische und polare Koordinatensystem auf einen Blick [Kap. 3 L.Paul, 2008].	25
2.8. Umrechnung von kartesischen in polare Koordinaten und umgekehrt [Kap. 3 L.Paul, 2008].	26
2.9. Lokale Koordinaten vs. Weltkoordinaten [Kern, 2008].	27
2.10. Linke Grafik: Winkelzuordnung bei einer Rotation: α : Drehung Z-Achse. β : Drehung Y-Achse. γ : Drehung X-Achse Rechte Grafik: 90° Rotation um die Z-Achse [Faber, 2007].	28
3.1. Die Bewegungen für die Aufnahme eines 3D-Laserscans.	32
3.2. Überblick über den internen Aufbau der 3D-Einheit.	34
4.1. Hauptmodule der 3D-Erfassungssteuerung.	42
4.2. Das 3D-Synchronisations-Modul.	42
4.3. Protokollaufbau von einem 3D-Request und einem 3D-Acknowledge.	46
4.4. Paketaufbau von einem 3D-Datenprofil.	47
4.5. Der Harelautomat der 3D-Erfassungssteuerung.	53
4.6. Interner Ablauf der State-Machine der 3D-Erfassungssteuerung. . .	55

4.7. Der Aufbau einer Can-Nachricht an die Powercubeeinheit [vgl. Schunk GmbH, 2007a].	56
4.8. Der 3D-Single-Scan im Step-Modus (Aktivitätsdiagramm UML 2.0).	58
4.9. Die Rampenbewegung der Powercubedrehachse.	62
4.10. Die Datenermittlung während eines 2D-Laserscansvorganges im Interpolations-Modus.	63
4.11. Der Ablauf der Erfassungssteuerung auf dem C32 im Interpolations-Modus (Aktivitätsdiagramm UML 2.0).	64
4.12. Zeitliche Betrachtung der Datenermittlung und der Datenverarbeitung im Interpolations-Modus.	64
4.13. Ein Beispiel für eine mögliche Koordinierung mehrerer 3D-Single-Scans im Interpolation-Modus.	67
4.14. Ergebnis der 3D-Erfassung des Beispiels für den Interpolations-Modus.	68
4.15. Das lokale Koordinatensystem der 3D-Einheit.	70
4.16. Die Laserscannerausrichtung bei der 3D-Einheit.	71
4.17. Transformation in die Homeposition der Powercubedrehachse.	73
4.18. Die Translation von der Home-Position zur rechten, hinteren Gehäusesedecke.	74
5.1. Bild der GUI der Testanwendung	76
5.2. Visualisierung zweier 3D-Datenprofile, die im Step-Modus nacheinander ermittelt wurden.	81
5.3. Ist-Soll-Vergleich einer 3D-Erfassung mit einer Winkelauflösung von $1/4^\circ$	82
5.4. Links: Dauer der Schritte 2 bis 5 des Messvorgangs im Step-Modus. Rechts: Dauer der Schritte 3 bis 5 des Messvorgangs im Step-Modus.	84

1. Einführung

Die Nachbildung der Sicht des menschlichen Auges und damit die 3D-Anschauung faszinieren die Menschen seither. Immer wieder werden 3D-Anwendungen entwickelt bzw. entworfen mit dem Ziel, eine 3D-Ansicht zu erzeugen. Von der 3D-Brille, dem 3D-Poster bis zum 3D-Stadtmodell. Das Ziel ist eine realistische Betrachtung einer Umgebung oder eines Objektes. Ob die Darstellung virtuell oder real ist, spielt dabei keine Rolle. Der Mensch möchte sich ein Objekt realitätsnah vorstellen können und dazu eignet sich eine Ansicht in 3D hervorragend.

1.1. Motivation

3D hat längst in der Sensortechnik Fuß gefasst. In unterschiedlichen Branchen werden verschiedene Verfahren eingesetzt, um Objekte in 3D darzustellen. Bei Mercedes werden z.B. in der Produktion zur Qualitätssicherung „Messroboter“ eingesetzt. Sie haben die Aufgabe Werkstücke und Werkzeuge während des Herstellungsprozesses zu überprüfen. Mit Hilfe einer 3D-Erfassung wird ein „Soll-Ist-Vergleich“ des hergestellten Werkstücks gebildet. Dadurch lassen sich sowohl Fehler erkennen, als auch Daten für eine Optimierung des Produktionsablaufes gewinnen [vgl. Bertagnolli, 2004, S.801]. Ein weiteres Beispiel aus der Medizin ist die 3D-Aufnahme eines ungeborenen Kindes. Mit Ultraschall wird ein 3D-Bild des Kindes im Mutterleib aufgenommen, um eventuelle Herzfehler möglichst in einem frühen Entwicklungsstadium zu erkennen [vgl. Borowiack, 2006]. Wie die beiden Beispiele zeigen, nimmt die Bedeutung von 3D-Erfassungen von Jahr zu Jahr zu. Mit Fortschritt der Technik lassen sich immer mehr komplexere Objekte in 3D erfassen. So ist auch nicht weiter verwunderlich, dass gerade bei der Automatisierung von Industrieanlagen, 3D-Sensoren eine zunehmende Rolle spielen. Gefragt sind Sensortechniken, die schnell, zuverlässig und mit hoher Genauigkeit Eigenschaften wie Länge, Breite, Höhe, Tiefe oder Volumen eines Objektes ermitteln.

Für diese Art von Anforderungen eignen sich im Besonderen Lasermesssysteme, die auf Basis der optischen Pulslaufzeitmessung fungieren. Sie tasten berührungs-

los mit einer hohen Genauigkeit und Schnelligkeit die Oberfläche eines Objektes ab [vgl. Hugenschmidt, 2007, Kap.5].

Ein führender Hersteller von optischen Pulslaufzeitlaseremesssystemen ist der Sick-Konzern [s. Sick AG, 2007] mit Hauptsitz in Waldkirch, Baden-Württemberg. Zum Konzern gehört die Tochter Sick Ibeo GmbH (nachfolgend nur mit Ibeo bezeichnet) aus Hamburg, die seit Jahren 2D-Laserscanner auf Basis optischer Pulslaufzeitmessung produzieren.

Anfang 2007 stellte die Firma Lase GmbH [s. Lase-GmbH, 2007], die vorwiegend Laserscanner der Firma Ibeo einsetzen, um Automatisierungsanwendungen zu entwickeln, eine Speziallösung für die Hamburger Hafen Lagerhaus AG zum Vermessen und Positionieren von Containern vor. In dieser Speziallösung wurde eine 3D-Anwendung über einen externen PC realisiert, welche einen 2D-Laserscanner und eine Motorservoschwenkeinheit namens Powercube (nachfolgend nur mit Powercubeinheit bezeichnet) verwendet.

Mitte 2007 entschloss sich die Firma Ibeo den Hardwareaufbau von diesem „3D-Laserscanner“ der Firma Lase zu übernehmen, um einen eigenen 3D-Laserscanner zu entwickeln. Wie in Abb. 1.1 zu sehen ist, werden im Unterschied zur Lösung der Lase GmbH, der Laserscanner und die Powercubeinheit zu einer logischen 3D-Einheit „verheiratet“, anstatt sie über einen externen PC zu synchronisieren. Kommuniziert wird nur über eine Datenschnittstelle. Durch diese Trennung von Anwendung und 3D-Datenerfassung ergeben sich folgende Vor- und Nachteile:

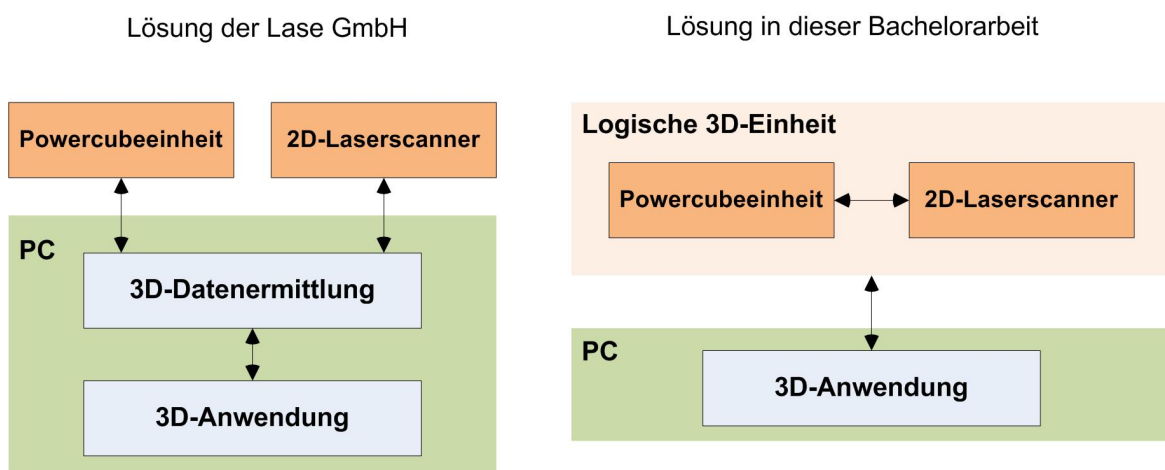


Abbildung 1.1.: Vergleich des 3D-Laserscanners der Firma Lase und dieser Bachelorarbeit.

Vorteile:

1. **Vereinfachung der Datenschnittstelle für den Anwender der 3D-Einheit:**
Für einen Anwender der 3D-Einheit, z.B. die Lase GmbH, würde sich die Datenschnittstelle deutlich vereinfachen. Eine Synchronisation und somit eine komplexe Bedienung der beiden Einheiten würde für ihn entfallen. Der Anwender würde „fertige“, vorbehandelte 3D-Daten von der neuen 3D-Einheit erhalten, die er gleich weiterverarbeiten könnte.
2. **Verbessertes Handling bei der 3D-Erfassung:**
Das Handling der 3D-Erfassung der logische 3D-Einheit ist weniger komplex als z.B. bei der Lösung der Lase GmbH. Zeitkritische Abläufe bleiben durch kurze Kommunikationswege auf die innere Architektur der 3D-Einheit beschränkt und sind somit überschaubarer. Echtzeitalgorithmen ließen sich leichter umsetzen. Die Fehleranfälligkeit durch äußere Einflüsse würde abnehmen und ein Fehler würde leichter lokalisierbar sein, als wenn beide Einheiten über verschiedene Schnittstellen bedient werden müssten. Gerade bei Outdooranwendungen sind die eben geschilderten Eigenschaften ein enormer Vorteil, da es bei Anwendungen im Freien oft zu unvorhersehbaren Szenarien kommen kann.

Nachteile:

1. **Eventuelle erschwerte Fehlerbehandlung:**
Kommt es zu einem Fehler bei der 3D-Erfassung in der 3D-Einheit, könnte eine Fehlerbehandlung je nach Art des Fehlers vom Kunden nicht mehr selbst durchgeführt werden. Support vor Ort müsste organisiert werden oder die 3D-Einheit müsste gegebenenfalls zum Hersteller zur Reparatur geschickt werden.
2. **Benutzer ist angewiesen auf Funktionalität der 3D-Einheit:**
Der Anwender ist völlig angewiesen auf die Funktionalität der 3D-Einheit. Er kann selber keine Konzepte für Synchronisation bieten bzw. erstellen.

In meiner Bachelorarbeit wird diese neue 3D-Einheit charakterisiert und ein Prototyp entworfen.

1.2. Einsatzgebiete und verwandte Lösungen

Wie in der Angebotsübersicht unter [s. Directindustry, 2007] zu sehen ist, existieren 3D-Laserscanner in verschiedenen Varianten. Die meisten angebotenen 3D-Laserscanner operieren im Nahbereich bis 3 m und führen eine Feinabtastung der Messobjekte durch. Oft verwendete Techniken sind Triangulation [s. Gühning, 2002] oder Laser-Interferometrie [s. Hugenschmidt, 2007, Kap.5]. Anwendungen sind z.B. Produktionsstätten, wo Objekte in hoher Auflösung erkannt werden müssen oder in Logistikunternehmen, wo die Größe der Pakete mit Hilfe von Laserscannern ermittelt werden.

Der Laserscanner, der in der 3D-Einheit dieser Bachelorarbeit verwendet wird, ist ein Distanzlaserscanner, der nach dem Time-of-Flight-Prinzip (Pulslaufzeitmessung) [vgl. Hugenschmidt, 2007, Kap.5] arbeitet. Typische Anwendungen befinden sich vor allem im Mittel- und Fernbereich, wo Distanzen zwischen 20 m und 250 m überbrückt werden müssen. Folgende Anwendungsfälle wären möglich:

- Füllstandmessung von Silos (z.B. Getreide oder Weizen).
- Volumenmessungen von Müll zur Weiterverarbeitung in Deponien.
- Volumenmessungen im Untertageabbau von Kohle und Eisen (Berechnung der abgebauten Masse).
- Vermessung von kleinen Schiffen bei Schleusungen.
- Materialvermessung in Lagerhallen.

In diesem Anwendungsbereich gibt es neben dem Sick-Konzern nur eine Hand voll anderer Anbieter bzw. Hersteller, wie z.B. die Firmen Riegl [Riegl, 2007] und Faro Europe [Faro Europe, 2007], die Lasermesssysteme in diesem Anwendungsbereich offerieren. Daneben gibt es noch einige Firmen wie die bereits erwähnte Firma Lase GmbH, die Laserscanner dieser Hersteller einsetzen, um eigene 3D-Anwendungen zu realisieren.

1.3. Zielsetzung und Abgrenzung

Diese Bachelorarbeit beschäftigt sich mit der Zusammenführung des 2D-Laserscanners und der Powercubeeinheit zu einer logischen 3D-Einheit bzw. zu einem logischen 3D-Laserscanner. Um den logischen 3D-Laserscanner bilden zu können, müssen die Powercubeeinheit und der 2D-Laserscanner intern auf einen Mikrokontroller „verheiratet“ werden. Die „Verheiratung“ beinhaltet, wie in Abb. 1.2 zu sehen, eine Synchronisation zwischen dem Laserscanner und der Powercubeeinheit, eine Datenvorbehandlung und eine Anwenderschnittstelle, die mit dem Benutzer der 3D-Einheit kommunizieren kann. Für diese Aufgaben müssen die Eigenschaften, Methoden und Algorithmen charakterisiert und entwickelt werden. Eine Weiterverarbeitung der 3D-Daten, wie z.B. die Mustererkennung, wird in dieser Arbeit nicht behandelt.

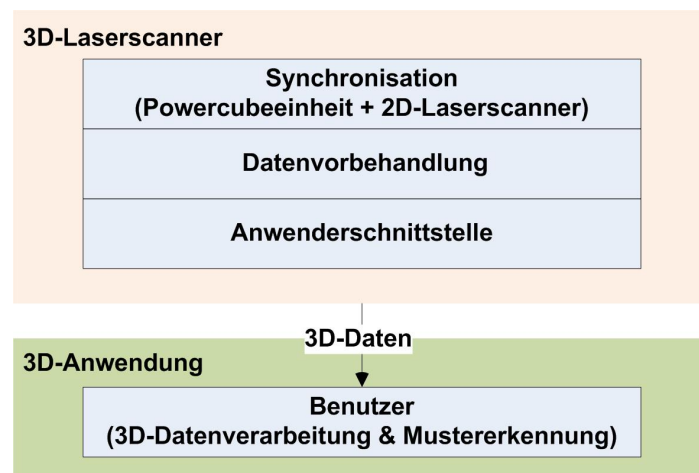


Abbildung 1.2.: Abstraktes Schichtmodell der Verarbeitung von 3D-Daten.

1.4. Gliederung der Arbeit

Die Bachelorarbeit gliedert sich in folgende Kapitel:

- **Kapitel 2:**
Die Einordnung der 3D-Messtechniken, Grundlagen der Lasermesstechnik und der 3D-Koordinatenverarbeitung werden erläutert.
- **Kapitel 3:**
Die Rahmenbedingungen werden für die Entwicklung der 3D-Einheit dargestellt. Es werden die Eigenschaften der 3D-Einheit und ihrer beiden Komponenten (Powercubeeinheit und Laserscanner) näher erläutert. Weiter werden hardware- und softwaretechnische Vorgaben der 3D-Einheit besprochen und die resultierenden Anforderungen für eine Entwicklung der logischen 3D-Einheit dargestellt.
- **Kapitel 4:**
Das Design und die Realisierung werden ausführlich beschrieben. Es wird gezeigt, wie die 3D-Erfassung realisiert wurde und welche Module dafür entwickelt wurden.
- **Kapitel 5:**
Die Qualitätssicherung der Entwicklung der 3D-Einheit wird betrachtet.
- **Kapitel 6:**
Am Ende wird ein Fazit und Ausblick gegeben. Es wird dargestellt, was erreicht wurde, welche Punkte offen sind und welche Verbesserungen und Erweiterungen möglich sind.

2. Grundlagen

In diesem Kapitel werden die verschiedenen 3D-Messtechniken eingeordnet. Anschließend werden die Grundlagen der Lasermesstechnik erläutert, die für die 3D-Erfassung dieser Bachelorarbeit von Bedeutung waren. Am Schluss werden die mathematischen Grundlagen und Grundbegriffe zur Koordinatenverarbeitung dargestellt, die bei der Verarbeitung der 3D-Daten eine entscheidene Rolle spielen.

2.1. Kurzüberblick und Einordnung von 3D-Messmethoden

Für eine Erfassung von 3D-Daten stehen bereits mehrere Varianten von Messmethoden zur Verfügung, die in verschiedenen Anwendungsgebieten benutzt werden. Hierbei ist darauf hinzuweisen, dass die Messmethoden auch bei 1D- und 2D-Messungen eingesetzt werden. Wie aus einer 1D- am Ende eine 3D-Messung wird, wird unter Punkt 2.2.2 am Beispiel eines Laserscanners näher erläutert. Nachfolgend werden kurz einige verschiedene 3D-Messmethoden oberflächlich dargestellt bzw. eingeordnet. Genauere Beschreibungen bzw. die Funktionsweise dieser Methoden oder ihrer abgeleiteten Verfahren werden in dieser Arbeit nicht näher betrachtet. Referenzen zu den Methoden werden während ihrer Betrachtung gegeben.

2.1.1. Passive Messmethoden

Die Messmethoden werden prinzipiell in passive und aktive Messmethoden unterteilt. Passive Messmethoden haben keine eigene Energiequelle, z.B. eine Lichtquelle, um ein Objekt erfassen zu können. Für die Messung benutzen sie die natürlichen Quellen der Umgebung. Sie sind abhängig von dem Strahlungsabfall der

Objekte. Beispiele für eine passive Messmethode sind das Stereosehen oder das Schattierungsverfahren [vgl. Kap.2 Elkhaili, 2005].

2.1.2. Aktive Messmethoden

Aktive Messmethoden greifen im Gegensatz zu den passiven Methoden direkt in die „Messszene“ ein. Dies kann mit Hilfe von Strahlen oder direkter Berührung passieren. Abbildung 2.1 zeigt eine kurze Übersicht über verschiedene Messmethoden der aktiven 3D-Erfassung. Die aktiven 3D-Messmethoden werden unterteilt in taktile und berührungslose Methoden.

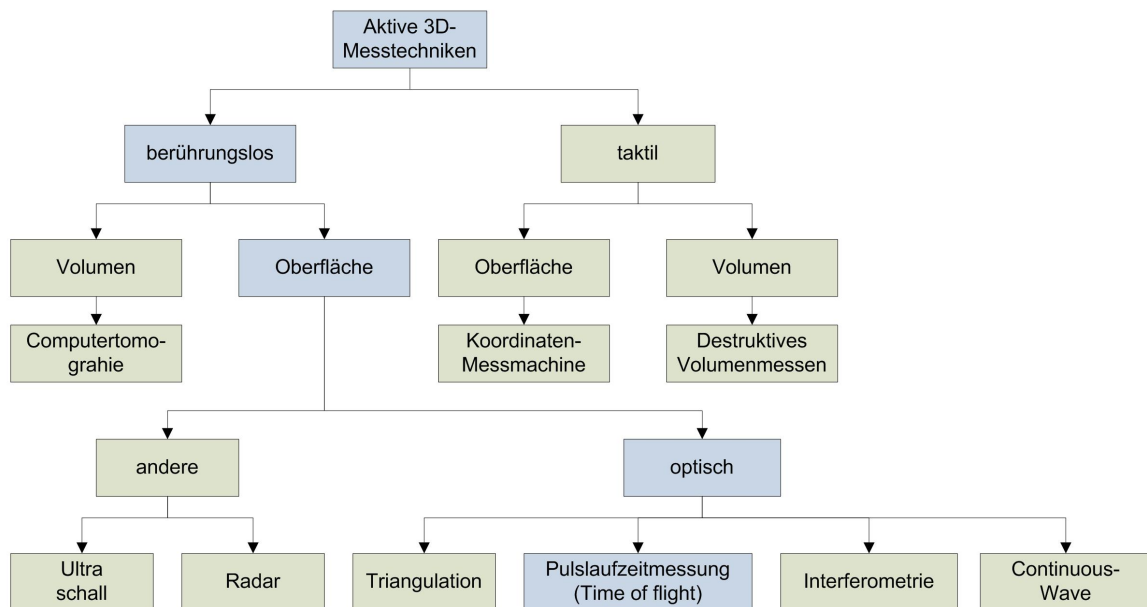


Abbildung 2.1.: Überblick über die aktiven Messmethoden [vgl. Kap.2 Gühring, 2002].

Taktile Messmethoden

Taktil bedeutet tasten bzw. berühren. Ein Objekt wird bei diesen Methoden „angefasst“. Ein Beispiel für eine taktile Messmethode ist die Koordinatenmessmaschine. Ein Tastkopf fährt hier die einzelnen Punkte des Objektes ab. Wird das Objekt vom Tastkopf berührt, wird eine Messung angestoßen und die Position des Tastkopfes wird ermittelt. An Hand dieser gewonnenen Positionen kann die Oberfläche des Objektes rekonstruiert werden [vgl. Kap.2 Gühring, 2002].

Berührungslose Messmethoden

Diese Methoden tasten die Messobjekte berührungslos ab. Es findet kein körperlicher Kontakt zwischen dem Messgerät und dem Messobjekt statt. Das Messobjekt wird gezielt mit Energie bestrahlt. Die reflektierte oder übertragene Energie wird von einem Detektor wieder aufgefangen und verarbeitet [vgl. Kap. 2.1.2 Elkhaili, 2005].

Beispiele für berührungslose Messsysteme sind Computertomografie (Röntgenstrahlung), Radar (Funkwellen), Ultraschall (Schallwellen) und Laserscanner (Lichtstrahlen). Die Laserscanner gehören zu den optischen Messsystemen. Häufig eingesetzte Verfahren bei optischen Messsystemen sind Interferometrie, Triangulation, Continuous Wave (CW) und die Pulslaufzeitmessung (englisch Time-of-flight kurz TOF). Informationen über die verschiedenen Messsysteme bzw. ihrer Verfahren sind unter anderem unter den Quellen [Elkhaili, 2005], [Hugenschmidt, 2007], [Gühring, 2002] zu finden.

Optische Messsysteme mit den Verfahren Interferometrie und Triangulation werden häufig im Nahbereich bis 3m eingesetzt. CW- und TOF-Laserscanner fangen dagegen etwa bei 1 Meter an und haben eine Reichweite von 200m bis 1 km. Beide Verfahren zeichnen sich durch geringe Abweichungen im mm-Bereich bei größeren Entfernungen aus [vgl. Hugenschmidt, 2007, Kap.5].

2.2. Messen mit einem Laserscanner

2.2.1. Grundbegriffe

Remissionen:

Die reflektierte Lichtenergie eines Messobjektes wird in der Physik als Remission bezeichnet. Oft wird auch der Begriff „Echo“ verwendet.

Die Definition 100 % Reflexion:

Als 100 % Reflexion wird ein Empfang eines Signals einer perfekt diffusen reflektierenden weißen Oberfläche definiert. An Hand dieser Definition ergeben sich auch Reflexionen, die größer als 100 % sind. Darunter fallen z.B. spiegelnde Oberflächen oder spezielle Reflektoren, die das Licht gerichtet reflektieren [vgl. Sick Ibeo GmbH, 2004a, Kap.3].

Reflektionsgrad:

Der Reflektionsgrad ist das Verhältnis zwischen einfallender und reflektierter Lichtenergie. Bei einem Verhältnis von 1, wurde die gesamte Lichtenergie von einem Objekt reflektiert.

Einordnung des Begriffes „Lasermessung“ und „Laserscan“:

Der Begriff Lasermessung wird oft in der Anwendung in verschiedenen Zusammenhängen verwendet. Mit einer Lasermessung kann die Ermittlung eines Messpunktes oder auch die Ermittlung mehrerer Punkte für eine Fläche bzw. eines Raumes, gemeint sein. In dieser Arbeit wird der Begriff „Lasermessung“ benutzt, wenn ein Messpunkt erfasst wird. Für eine Erfassung mehrerer Punkte wird der Begriff „Laserscan“ verwendet.

Interlace-Messen:

Der Begriff „Interlace“ kommt aus der Bildverarbeitung und bedeutet zu Deutsch „Zeilensprungverfahren“. Ein Bild wird dabei aus zwei oder mehr Halbbildern erstellt (s. Abb. 2.2).

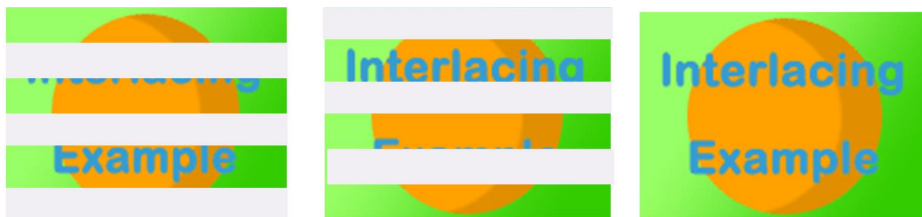


Abbildung 2.2.: Links: Zeichnung des ersten Halbbildes. Mitte: Zeichnung des zweiten Halbbildes. Rechts: Fertig gezeichnetes Bild [Wikipedia.org, 2008b].

In der Lasermesstechnik kommt diese Technik oft zum Einsatz, wenn das Messgerät an die Grenzen seiner Technik stößt. Sie wird benutzt, um die Winkelauflösung zu erhöhen. Aufgrund der Bauart des Lasersensors bzw. seines Messverfahrens muss zwischen zwei Messungen eine bestimmte Winkelschrittweite eingehalten werden. Um diese Winkelschritte „künstlich“ zu verkleinern und damit die Auflösungen einer gesamten Messaufnahme zu erhöhen, werden prinzipiell zwei oder mehrere Laserscans aus verschiedenen Startwinkeln durchgeführt. Diese Lasers-

canneraufnahmen werden dann übereinander gelegt und ergeben ein neues Bild mit einer erhöhten Winkelauflösung.

2.2.2. Das Prinzip der 3D-Erfassung mit einem Laserscanner

Im Folgenden wird veranschaulicht, wie eine 3D-Erfassung mit Hilfe eines Laserscanners entsteht. Ein 3D-Raum wird in 3 Achsen gegliedert. Das bedeutet 3 Schritte sind notwendig, um eine 3D-Messung durchzuführen. Welche Achse, welche Bedeutung erhält, ist dabei beliebig. In dem folgenden Beispiel haben die Achsen folgende Bedeutungen:

- Die X-Achse ist die Tiefen- bzw. die Distanzachse.
- Die Y-Achse ist die Längenangachse.
- Die Z-Achse ist die Breitenachse.

Der Ursprung einer 3D-Erfassung ist die 1D-Erfassung. Bei einer 1D-Erfassung wird mit Hilfe eines Laserscanners die Distanz/Tiefe eines Punktes gemessen. Wird der Laserscanner im zweiten Schritt von oben nach unten geschwenkt, so entsteht eine 2D-Messung. Die Daten einer 2D-Erfassung bilden eine Linie ab, die aus mehreren Punkten besteht, wobei jeder Punkt eine andere Distanz zum Laserscanner besitzen kann (s. Abb. 2.3). Wird der Laserscanner im dritten Schritt von rechts nach links gedreht, kann eine Fläche gemessen werden. Finden diese Schwenkungen und Drehungen gleichzeitig statt, entstehen „schräge“ 2D-Punktlinien (s. Abb. 2.4). Bei abwechselten Schwenkungen und Drehungen werden die Daten senkrechter 2D-Punktlinien erfasst (s. Abb. 2.4). Aus mehreren 2D-Punktlinien kann anschließend eine Messumgebung in 3D dargestellt werden.

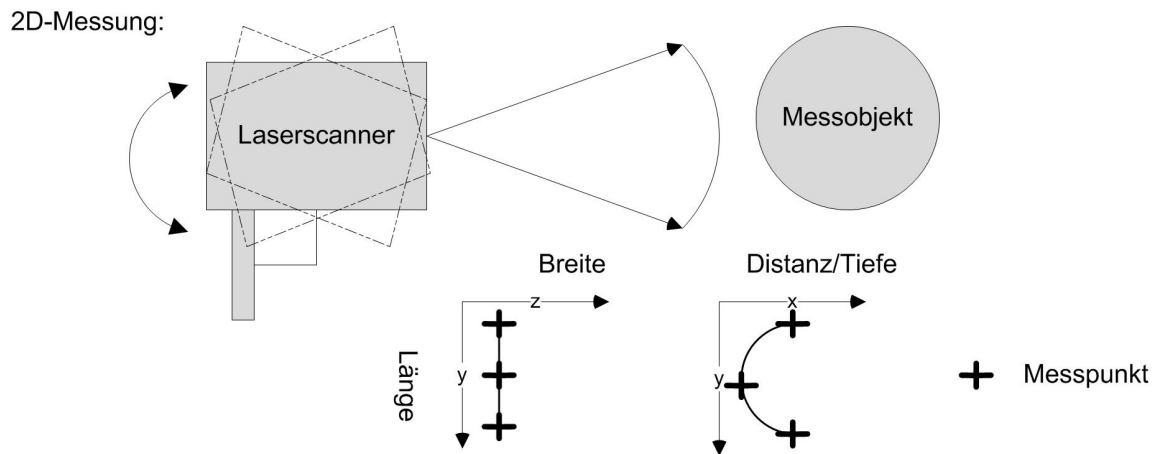


Abbildung 2.3.: 2D-Messung einer Punktlinie.

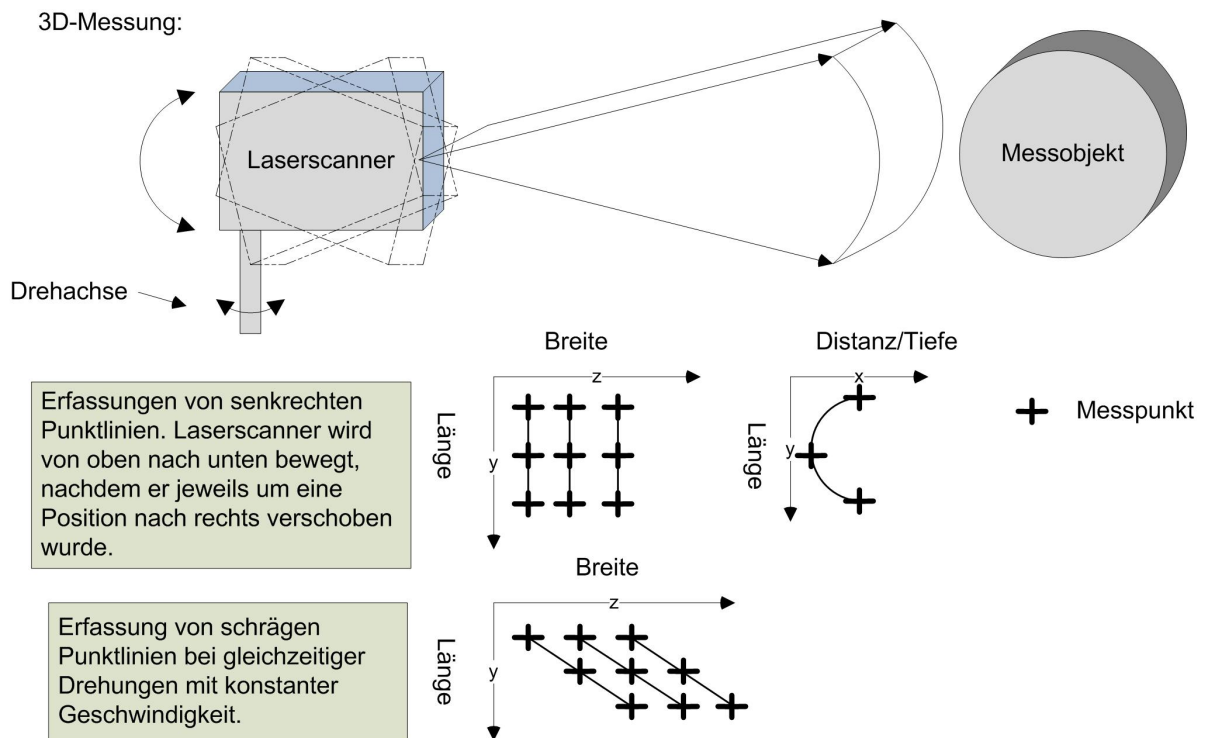


Abbildung 2.4.: 3D-Messung, bestehend aus mehreren 2D-Punktlinien.

2.2.3. Einflüsse auf die Entfernungsmessung beim Laserscannen

Die Qualität der Entfernungsmessung bei optischen Lasersystemen ist zum einen abhängig von der Art des Lasersensors und von der Art der Messtechnik (Interferometrie, Triangulation, Continuous Wave(CW) und Pulslaufzeitmessung(TOF)). Auf die verschiedenen Laserarten wird in dieser Arbeit nicht eingegangen. Für mehr Informationen bzw. Einblick über die verschiedenen Laserarten [s. Kap.9 Eichler, 2007].

Andere Abhängigkeiten entstehen durch die Oberfläche des Messobjektes selber. Faktoren sind der Reflektionswinkel und der Reflektionsgrad, die je nach Art der Oberfläche der Messobjekte mal schlechter und mal besser ausfallen. Je besser diese beiden Faktoren, desto mehr Remissionen können empfangen werden und desto höher ist die Reichweite, in der ein Objekt noch erfasst werden kann. Die Faktoren sind abhängig von Helligkeit und Glätte der Oberfläche. Zusammengefasst ergeben sich folgende Regeln für den Reflektionswinkel und dem Reflektionsgrad in Abhängigkeit der Oberfläche:

- Typischerweise sind die meisten Oberflächen eher rau als glatt. Raue Oberflächen fördern eine diffuse Reflektierung (s. Abb. 2.5). Je diffuser eine Reflektierung, desto höher die Wahrscheinlichkeit eines schlechten Reflektionswinkels.
- Senkrecht Laserbestrahlen einer rauen Oberfläche erhöht die Wahrscheinlichkeit eines guten Reflektionswinkels (s. Abb. 2.6).
- Je dunkler eine Oberfläche, desto schlechter der Reflektionsgrad und um so weniger Remissionen werden reflektiert.
- Materialien wie Kunststoffreflektoren oder Reflektionsfolie ermöglichen einen Reflektionsgrad über eins bzw. eine Reflektion über den definierten 100%. Die Lichtstrahlen werden dabei gerichtet in die Herkunftsrichtung zum Laserscanner zurück reflektiert.
- Prinzipiell je schlechter der Reflektionswinkel, desto schlechter der Reflektionsgrad.

Die Tabelle 2.1 zeigt exemplarisch typische Remissionswerte einiger Materialien. Für ausführliche Informationen und Vergleiche über die Einflüsse auf die Entfernungsmessung s. [Kap.3 Sick Ibeo GmbH, 2004a].

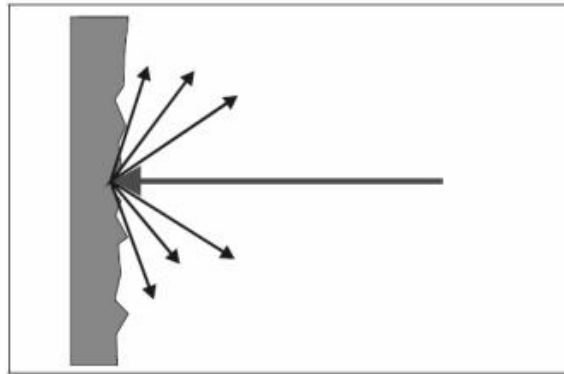


Abbildung 2.5.: Diffuse Reflexion einer rauhen Oberfläche
[Kap.3 Sick Ibeo GmbH, 2004a].

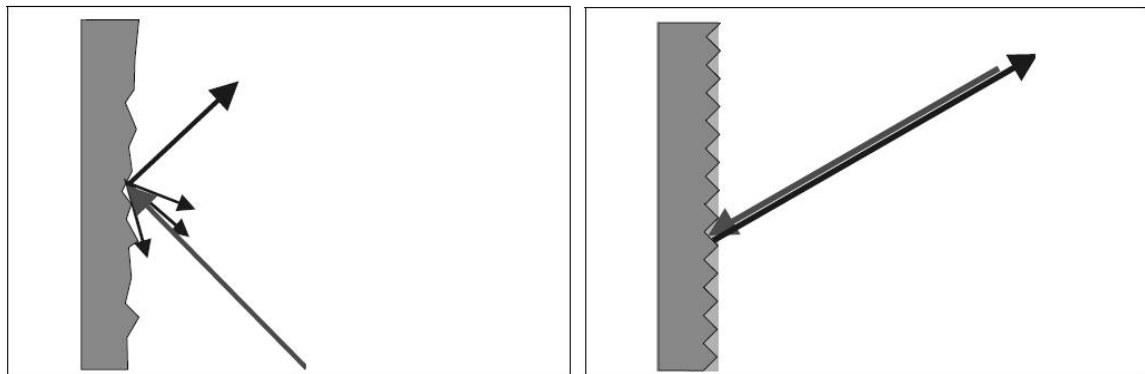


Abbildung 2.6.: Links: schräges Laserbestrahlen \Rightarrow schlechter Reflektionswinkel.
Rechts: senkrecht Laserbestrahlen \Rightarrow guter Reflektionswinkel
[Kap.3 Sick Ibeo GmbH, 2004a].

Material	Remission in %
schwarzer Autolack, matt	5
schwarzer Photokarton, matt	10
grauer Beton	18
weißer Karton	90
weißer Gips	100
Reflektionsfolie	>300

Tabelle 2.1.: Remissionswerte in Abhängigkeit von der Art der Oberfläche
[Kap. 3 Sick Ibeo GmbH, 2004a].

2.3. Grundlagen der 3D-Koordinatenverarbeitung

2.3.1. Polar vs. Kartesisch

Die Objekte in einer 3D-Umgebung werden durch zusammenhängende Punkte in einem Koordinatensystem beschrieben. Generell können Koordinaten im polaren oder kartesischen Format dargestellt werden. Ein Punkt im kartesischen Koordinatensystem wird durch eine Position auf der X-, Y- und Z-Achse beschrieben. Dieses Format wird in der Bildverarbeitung am meisten verwendet. Dagegen wird im polaren Koordinatensystem ein Punkt durch den Abstand zum Ursprung (Radialkoordinate), und den Winkeln (Winkelkoordinate) zwischen den Achsen beschrieben (s. Abb. 2.7). Polare Koordinaten werden auch oft als Kugelkoordinaten bezeichnet.

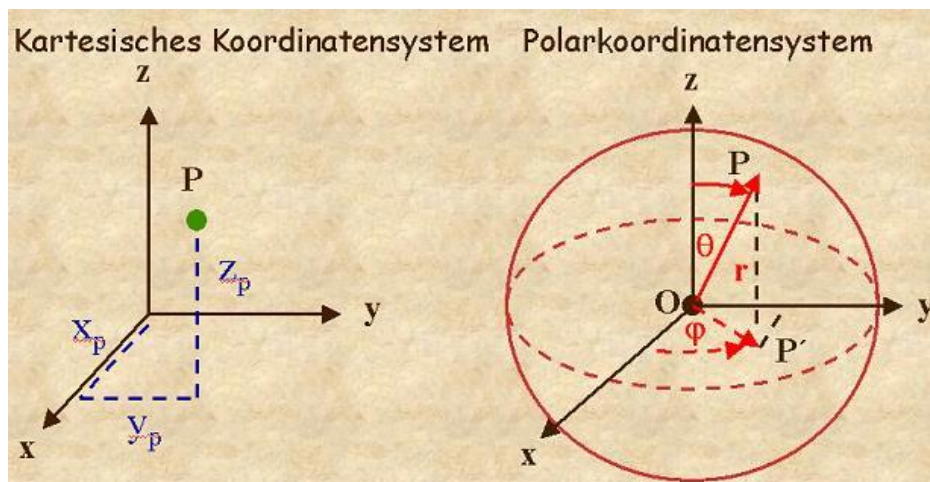


Abbildung 2.7.: Das kartesische und polare Koordinatensystem auf einen Blick [Kap. 3 L.Paul, 2008].

Die Abb. 2.8 zeigt, wie mit geometrischen Funktionen ein 3D-Punkt vom kartesischen in das polare Koordinatensystem und umgekehrt transformiert werden kann.

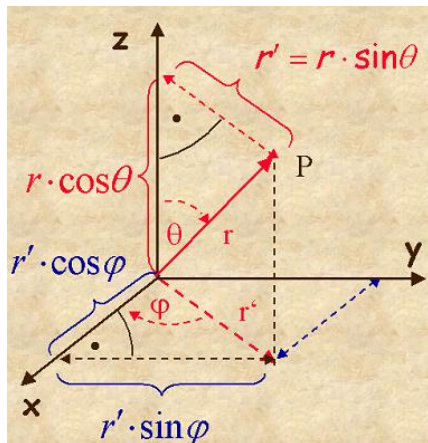


Abbildung 2.8.: Umrechnung von kartesischen in polare Koordinaten und umgekehrt [Kap. 3 L.Paul, 2008].

3D-Kartesisch zu 3D-Polar

$$xp = r \cdot \sin(\Theta) \cdot \cos(\phi)$$

$$yp = r \cdot \sin(\Theta) \cdot \sin(\phi)$$

$$zp = r \cdot \cos(\Theta)$$

3D-Polar zu 3D-Kartesisch

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

$$\Theta = \frac{zp}{r}$$

$$\phi = \arctan \frac{yp}{xp}$$

2.3.2. Lokales vs. weltliches Koordinatensystem

In der Verarbeitung von 3D-Daten wird zwischen zwei Betrachtungsarten der Koordinatensysteme unterschieden. Auf der einen Seite gibt es die Betrachtung aus dem globalen weltlichen Koordinatensystem und auf der anderen Seite die Betrachtung aus dem lokalen Koordinatensystem.

Lokales Koordinatensystem

Der Ursprung des lokalen Koordinatensystems kann ein Punkt des Messobjektes sein. Es wird daher auch häufig von „aus der Sicht des Messobjektes“ gesprochen. Mit Hilfe des lokalen Koordinatensystems können dann die Länge, Breite, Höhe und das Volumen ermittelt werden. Bei einem Laserscanner ist der Ursprung des lokalen Koordinatensystems oft die Mitte des Laserkopfes.

Globales Weltkoordinatensystem

Das Weltkoordinatensystem bezieht sich auf die komplette Messumgebung. Der Ursprung des Weltkoordinatensystems ist daher in der Regel ein definierter fester Punkt in der Messumgebung. Dies kann z.B. eine Ecke im Raum sein. Der Standort des Ursprungs des Weltkoordinatensystems bleibt im Gegensatz zum Ursprung des lokalen Koordinatensystems bei einer Messung immer an einem festen Punkt. Die Abbildung 2.9 zeigt den Unterschied zwischen einem lokalen und weltlichen Koordinatensystem. Zur Unterscheidung sind die Achsen beim Weltkoordinatensystem groß geschrieben.

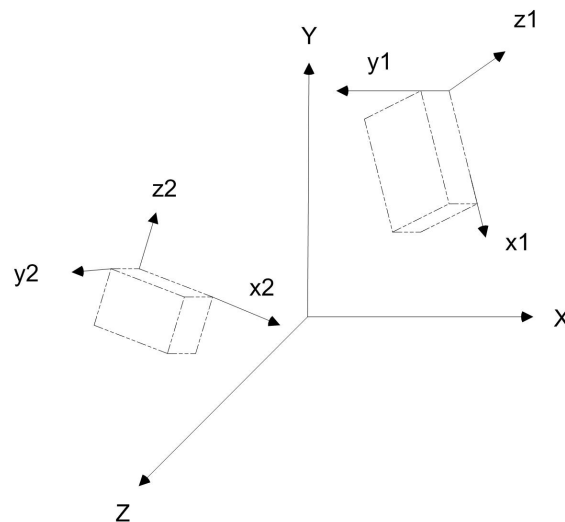


Abbildung 2.9.: Lokale Koordinaten vs. Weltkoordinaten [Kern, 2008].

2.3.3. Koordinatentransformationen

Ein Messgerät kennt in der Regel bei der Aufnahme nur seine eigenen lokalen Koordinaten. Die Messdaten des Aufnahmegerätes sind relativ zum Koordinatenursprung des Messgerätes. Um diese Daten mit anderen Messungen zu vergleichen oder einfach nur ihre Lokalität im Raum feststellen zu können, müssen die Daten zu einem anderen Bezugspunkt transformiert werden. Dazu werden die Koordinaten von dem einen in das andere Koordinatensystem übertragen. Diese Übertragung wird Koordinatentransformation genannt. In Abbildung 2.9 sind zwei Objekte zu sehen. Um z.B. zu ermitteln, ob die Objekte gleich groß sind, müssen die Daten in das weltliche Koordinatensystem übersetzt werden. Die mathematische Betrachtung der folgenden Basistransformation erfolgt mit homogenen Koordinaten. In Verbindung mit der Matrixschreibweise fördern homogene Koordinaten das Verständnis von Transformation und ihrer Verarbeitung [vgl. Kap.4 Grieger, 1992].

Translation

Unter einer Translation wird eine Verschiebung des Koordinatensystems entlang einer oder mehrerer Achsen verstanden.

Mathematisch lässt sich ein Punkt $P=(x,y,z)$ verschoben um den Vector (tx,ty,tz)

$(x',y',z') := (x+tx,y+ty,z+tz)$

in homogenen Koordinaten folgendermaßen ausdrücken:

$$\begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & tx \\ 0 & 1 & 0 & ty \\ 0 & 0 & 1 & tz \\ 0 & 0 & 0 & 1 \end{pmatrix} * \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

Rotation

Eine Rotation eines Koordinatensystems ist eine Drehung um eine oder mehrere Achsen des Koordinatensystems. Das Zentrum der Rotation ist immer der Ursprung des Koordinatensystems. Die Winkel sind entgegen dem Uhrzeiger positiv definiert. Die linke Grafik der Abbildung 2.10 zeigt die Winkelzuordnung bei einer Rotation, während bei der rechten Grafik der Abbildung 2.10 exemplarisch eine 90° Rotation um die Z-Achse dargestellt wird. Die Tabelle 2.2 stellt wiederum einen Überblick der mathematischen Berechnungen für eine Rotation um X-, Y- und Z-Achse dar.

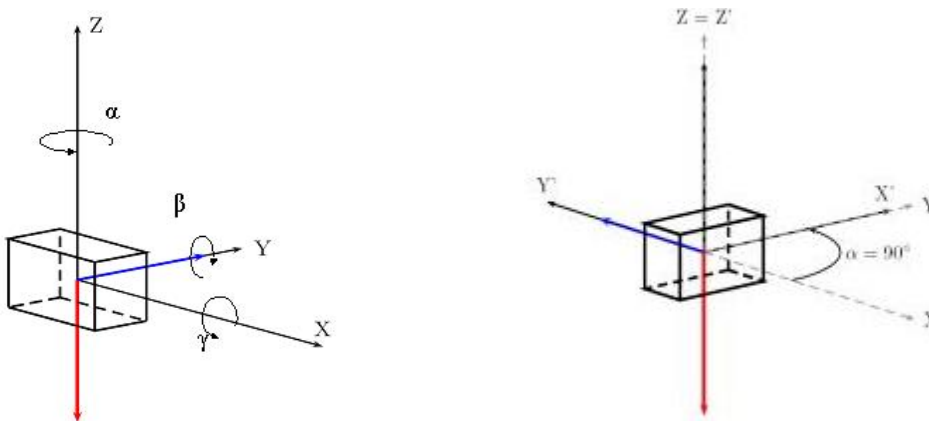


Abbildung 2.10.: Linke Grafik: Winkelzuordnung bei einer Rotation:

α : Drehung Z-Achse. β : Drehung Y-Achse. γ : Drehung X-Achse

Rechte Grafik: 90° Rotation um die Z-Achse [Faber, 2007].

Rotation	Ausdruck als Zuweisung	mit homogenen Koordinaten
Z-Achse	$x' := x \cdot \cos(\alpha) - y \cdot \sin(\alpha)$ $y' := x \cdot \sin(\alpha) + y \cdot \cos(\alpha)$ $z' := z$	$\begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} = \begin{pmatrix} \cos(\alpha) & -\sin(\alpha) & 0 & 0 \\ \sin(\alpha) & \cos(\alpha) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$
X-Achse	$x' := x$ $y' := y \cdot \cos(\gamma) - z \cdot \sin(\gamma)$ $z' := y \cdot \sin(\gamma) + z \cdot \cos(\gamma)$	$\begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & \cos(\gamma) & -\sin(\gamma) & 0 \\ 0 & \sin(\gamma) & \cos(\gamma) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$
Y-Achse	$x' := z \cdot \sin(\beta) + x \cdot \cos(\beta)$ $y' := y$ $z' := z \cdot \cos(\beta) - x \cdot \sin(\beta)$	$\begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} = \begin{pmatrix} \cos(\beta) & 0 & \sin(\beta) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(\beta) & 0 & \cos(\beta) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$

Tabelle 2.2.: Übersicht über die Rotationen um die verschiedenen Achsen.

2.4. Zusammenfassung

Es wurde gezeigt, dass in der 3D-Erfassungswelt verschiedene 3D-Messtechniken existieren, wobei die Wahl der Messtechnik abhängig davon ist, was erfasst werden soll. Weiterhin wurden die Begrifflichkeiten der Lasermesstechnik erläutert, die für die 3D-Erfassung dieser Bachelorarbeit wichtig sind und es wurde dargestellt, wie ein Laserscanner eine Messumgebung erfassen kann. In der Koordinatenverarbeitung wurde gezeigt, wie ein 3D-Messpunkt mit Hilfe von Transformationen verarbeitet werden kann und welche Dinge dabei beachtet werden müssen.

3. Rahmenbedingungen

In diesem Kapitel werden zuerst die beiden Einheiten, der Laserscanner und die Powercubeeinheit, der 3D-Einheit dargestellt. Anschließend wird gezeigt, mit welchen Bewegungen diese beiden Einheiten einen 3D-Laserscan erzeugen können und welche Grenzen dabei existieren. Nachfolgend werden einige Definitionen erläutert, die im weiteren Verlauf helfen, die 3D-Erfassung zu beschreiben. Im Anschluss daran wird der interne Aufbau der 3D-Einheit betrachtet. Es wird dargestellt, welche hardware- und softwaretechnischen Voraussetzungen für eine 3D-Erfassung bzw. einer logischen Verknüpfung der Powercubeeinheit und des Laserscanners existieren und welche Anforderungen daraus resultieren. Abschließend wird dargestellt, welche Randbedingungen bezogen auf Test und Entwicklungsumgebung, vorhanden sind.

3.1. Features der Akteure der 3D-Einheit

3.1.1. 2D-Laserscanner LRS-2100

Als 2D-Laserscanner für die 3D-Einheit ist der LRS-2100 von der Firma Sick-Ibeo GmbH vorgesehen. Folgende Auflistung zeigt die wichtigsten Features des LRS-2100:

- Die Erfassung der Distanzwerte erfolgt nach dem Prinzip der Pulslaufzeitmessung (TOF) (vgl. Kap. 2.1.1).
- Ein 360° kreisrotierender Laserkopf ermöglicht einen Laserscan in einem 300° Scanbereich (60° sind durch das Gehäuse abgedeckt). In einer Umdrehung wird ein 2D-Datenprofil erstellt [vgl. Kap.3 Sick Ibeo GmbH, 2004b].
- 3,9 mm genaue Messauflösung und 0,125° hohe Winkelauflösung [vgl. Kap.3 Sick Ibeo GmbH, 2004b].
- Reichweite: 250 m bei optimalen Verhältnissen (Reflektorfolie), 80 m bei 10% Remission [vgl. Kap.3 Sick Ibeo GmbH, 2004b].

3.1.2. Powercubeeinheit

Der zweite Akteur der 3D-Einheit ist die Powercubeeinheit. Der Name „Powercube“ steht eigentlich für eine ganze Produktpalette von Antriebsmodulen der Firma Schunk GmbH. Die richtige Bezeichnung der Powercubeeinheit lautet PDU-110-161. Die PDU-110-161 verfügt über einen Servomotor mit Präzisionsgetriebe, der über einen eigenen Controller angesteuert wird. Dieser Servomotor ermöglicht ein Fahren der Drehachse der Powercubeeinheit mit verschiedenen Bewegungsprofilen. Die Bewegungen der Drehachse werden weiter durch verschiedene Mechanismen, wie einer Bewegungskontrolle (Selbständiges Erkennen von Bewegungen über die eingestellten Systemgrenzen), einer Strom-, Spannungs- und Temperaturkontrolle überwacht. Kalibriert wird die Powercubeeinheit für eine „Homing-Funktionalität“. Diese ermöglicht unabhängig vom Status der Einheit eine Referenzfahrt in eine Homeposition. Die Homeposition (default bei 0°) ist eine feste Position, die durch einen Endschalter erkannt wird. So wird sichergestellt, dass zu jeder Zeit ein Referenzpunkt vorhanden ist, an dem sich die Bewegungsbefehle orientieren können [vgl. Schunk GmbH, 2007b, für Funktionen], [vgl. Schunk GmbH, 2008, für Technische Details], [vgl. Schunk GmbH, 2007a, für die Kommunikation].

3.2. Die Bewegung der 3D-Erfassung

Die Powercubeeinheit und der Laserscanner sind in einem Gehäuse verbaut, wobei der Laserscanner fest auf der Drehachse der Powercubeeinheit montiert ist.

Wie in Abb. 3.1 zu sehen, wird ein 2D-Laserscan durchgeführt, indem der Laserkopf um 360° um seine eigene Achse bewegt wird. Während dieser Umdrehung wird in festen Winkelabständen eine Distanzmessung durchgeführt. Diese 2D-Daten des Laserscanners werden zu 3D-Daten erweitert, indem die Powercubeeinheit den Laserscanner über seinen Motor von oben nach unten schwenkt. Das bedeutet, der Laserscanner führt aus verschiedenen Positionen 2D-Scans aus, die am Ende zu einem 3D-Laserscan zusammen gestellt werden. Durch die Art und Weise der 3D-Lasererfassung, könnte theoretisch ein kugelförmiges Scanfeld abgetastet werden, wobei der Radius der maximalen Reichweite des Laserscanners entsprechen würde.

Ein Teil des möglichen Sichtfeldes wird jedoch aufgrund des Aufbaus der 3D-Einheit verdeckt. Der Laserscanner kann maximal in einen Bereich zwischen 90° und 270° in einer 360° Drehung messen. Die Drehachse der Powercubeeinheit kann von der

festen 0° Position 45° nach links und rechts bewegt werden. Das entspricht einem Bewegungsbereich von 90°.

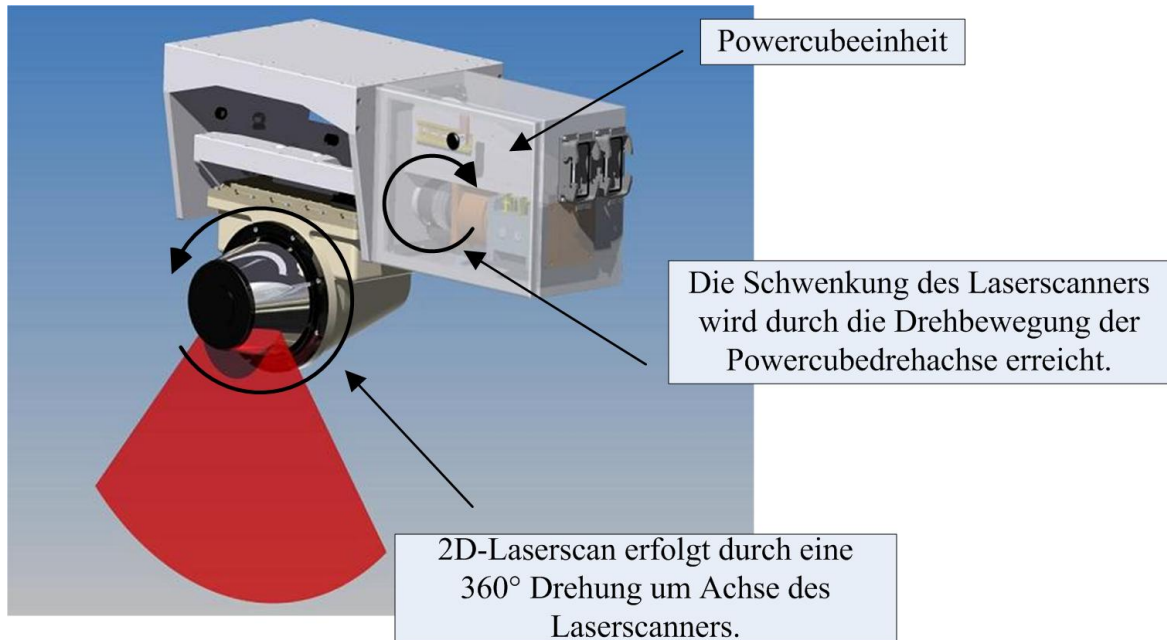


Abbildung 3.1.: Die Bewegungen für die Aufnahme eines 3D-Laserscans.

3.3. Definitionen für die 3D-Erfassung

Folgende Definitionen helfen beim Verständnis der Rahmenbedingungen, des Entwurfes und der Realisierung der 3D-Erfassung:

Definition 3D-Messpunkt

Ein 3D-Messpunkt ist eine Zusammensetzung einer 2D-Koordinate des 2D-Laserscanners und der zeitgleich aktuellen Position der Drehachse der Powercubeinheit. Weiter erhält ein 3D-Messpunkt einen Echo-Wert, der den Remissionswert zu einem Punkt darstellt.

Definition 3D-Datenprofil

Ein 3D-Datenprofil besteht aus mehreren 3D-Messpunkten, die aus einer Zusammenführung eines 2D-Profiles des Laserscanners und der jeweiligen zeitgleichen Positionsdaten der Drehachse der Powercubeeinheit entstanden sind.

Definition 3D-Single-Scan

Ein 3D-Single-Scan ist ein Messvorgang, wo ein 2D-Scan, also die Daten einer 360° Drehung des Kopfes des Laserscanners, mit den Positionsdaten der Drehachse der Powercubeeinheit zu einem 3D-Scan erweitert wurde und ein 3D-Datenprofil erstellt wurde.

Definition 3D-Komplett-Scan

Ein 3D-Komplett-Scan besteht aus einer Anzahl von 3D-Single-Scans, die in verschiedenen Winkelpositionen des konfigurierten Scanbereiches (maximal 90°) der Powercubeeinheit, während einer Bewegung von einer Start- bis zu einer Endposition der Powercubedrehachse, gemessen wurden.

3.4. Interner Aufbau der 3D-Einheit

3.4.1. Entwicklung einer 3D-Erfassungssteuerung auf dem C32

Die 3D-Erfassungssteuerung und damit die logische Verschmelzung der Powercubeeinheit und des Laserscanners dieser Bachelorarbeit wurden auf dem C32-Applikationsprozessor (nachfolgend nur mit C32 bezeichnet) des Laserscanners entwickelt. Der C32, bestehend aus dem 32-Bit fixed-point Prozessor-Modul TMS320C32-60 von Texas-Instrument, ist, wie in Abb. 3.2 zu sehen, einer von zwei Prozessoren des Laserscanners und ist vorgesehen für weiterverarbeitende Anwendungen der 2D-Laserdaten.

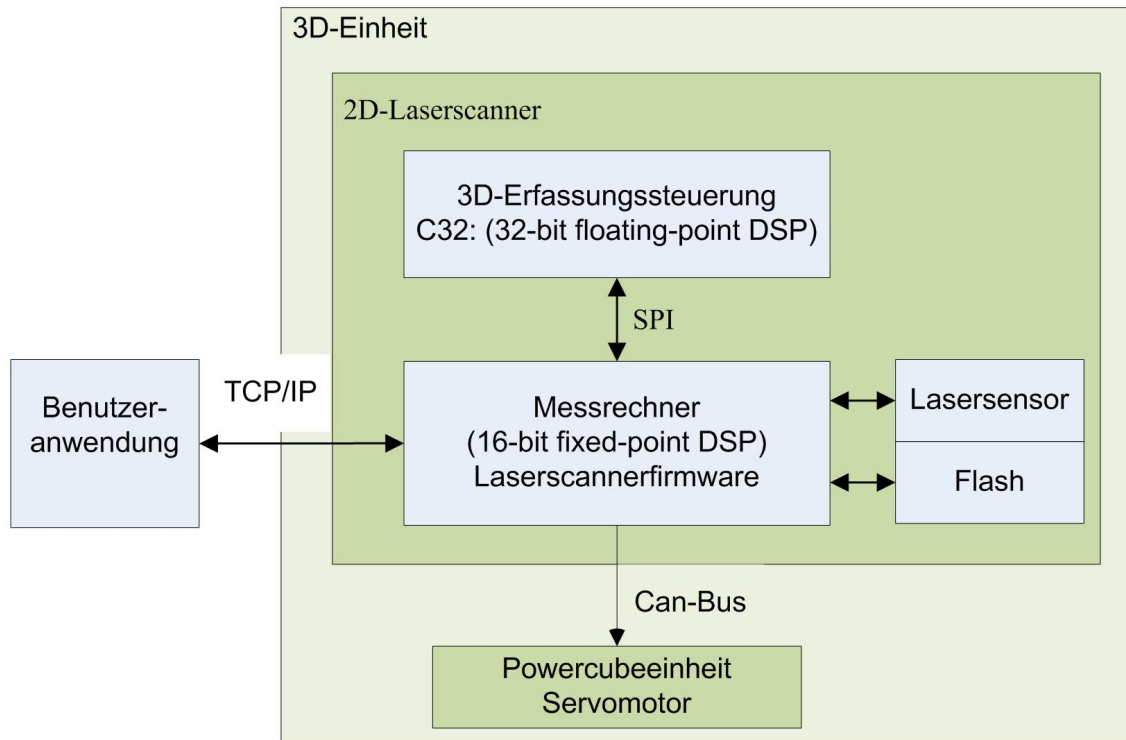


Abbildung 3.2.: Überblick über den internen Aufbau der 3D-Einheit.

Der zweite Prozessor wird Messrechner genannt. Der Messrechner ist der Hauptprozessor des Laserscanners und wird ebenfalls von Texas-Instrument hergestellt. Er hat folgende Eigenschaften:

- Setzt das 16-Bit fixed-point Prozessor-Modul „TMS320F243“ ein.
- Führt die Firmware des Laserscanners aus.
- Steuert alle Schnittstellen des Laserscanners.
- Zuständig für die 2D-Laserdatenerfassung bzw. der 2D-Lasersensorik.

3.4.2. Vorgaben für die Entwicklung auf dem C32

Vorgaben der Kommunikations-Software-Schnittstelle durch den Messrechner

Um einen 3D-Laserscan erzeugen zu können, wurden folgende Verbindungen festgelegt:

- 3D-Erfassungssteuerung \Leftrightarrow Powercubeeinheit.
- 3D-Erfassungssteuerung \Leftrightarrow Benutzer der 3D-Einheit.
- 3D-Erfassungssteuerung \Leftrightarrow Lasersensorik.

Wie in Abb. 3.2 zu sehen ist, sind die Powercubeeinheit über Can-Bus (Controller Area Network), eine Anwendung eines Benutzers der 3D-Einheit über TCP/IP und die 3D-Erfassungssteuerung über SPI (Serial Peripheral Interface) direkt am Messrechner angebunden. Das bedeutet, die komplette Kommunikation der 3D-Erfassungssteuerung geht über den Messrechner. Für eine Kommunikation über dem Messrechner sind folgende softwaretechnische Richtlinien vorgegeben:

- Kommuniziert wird mit dem Messrechner über die User-Service-Protocol-Schnittstelle (USP-Schnittstelle). Die USP-Schnittstelle besteht aus mehreren Funktionen, die in verschiedene Servicegruppen unterteilt werden [s. Kap.4 Sick Ibeo GmbH, 2006].
- Bei der Kommunikation zwischen der 3D-Erfassungssteuerung und dem Benutzer der 3D-Einheit über SPI und TCP/IP muss der Service „Application“ der USP-Schnittstelle und das User-Service-Protocol verwendet werden [s. Kap.4 und Kap.7.3.2 Sick Ibeo GmbH, 2006]. Die Datengröße für ein TCP/IP-Paket ist dabei mit 10240 Bytes spezifiziert.
- Die Kommunikation zwischen der 3D-Erfassungssteuerung und der Powercubeeinheit über SPI und Can-Bus kann über die USP-Schnittstelle konfiguriert werden, so dass kein User-Service-Protocol verwendet werden muss [s. Kap.5.5 Sick Ibeo GmbH, 2006].

Für einen transparenten Zugriff auf die USP-Schnittstelle und der Lasersensorik, wird der 3D-Erfassungssteuerung ein Application-Programming-Interface (API) „Basefunction“ bereit gestellt [vgl. Sick Ibeo GmbH, 2005]. Die Basefunction realisieren eine interrupt-basierte Spi-Kommunikation mit dem Messrechner.

Die Kommunikation zwischen einer Anwendung auf dem C32 über Spi/Can-Bus und über Spi/TCP/IP wurde bereits mehrfach in verschiedenen Projekten benutzt

bzw. getestet. Jedoch wurden beiden Kommunikationsarten bisher noch nie in einer Anwendung zusammen verwendet.

Kein IEEE-FORMAT

Der C32 auf dem die 3D-Erfassungssteuerung ausgeführt wird, unterstützt keine Standarddarstellungen nach dem IEEE-Standard. Sowohl der Standard für Fest- und Gleitkommazahlen [s. The IEEE, Inc, 1985] als auch der allgemeine Standard für Variablen in 32-Bit-Systemen wird nicht eingehalten. Texas Instruments hat für den C32 ein eigenes Float-Pointing-Format entworfen [s.Kap.5 Texas Instrument, 2004], stellt aber eine Assemblerfunktion zur Verfügung, mit dem eine Floating-Point-Zahl in das IEEE-Standard-Format formatiert werden kann. Alle anderen Variablen wie Char (normalerweise 1 Byte) und Short (2 Byte) sind im C32 4 Byte groß.

Zwischenspeicherung von 3D-Scandaten

Als Arbeitsspeicher wird der 3D-Erfassungssteuerung vom C32 256 KB bereit gestellt. Der Stack und die bisherigen globalen Variablen des Spi-Treibers, der Basefunction und der USP-Schnittstelle nehmen bereits ca. 61 KB des 256 KB großen Arbeitsspeichers ein (vgl. Anhang A). Aufgrund der Tatsache, dass jede Variable 4 Byte groß ist und unter der Annahme, dass ein 3D-Scan im maximalen Scanbereich der 3D-Einheit mit der maximalen Auflösung des Laserscanners statt findet, ergibt sich aus der in Tabelle 3.1 dargestellten Berechnung folgende Schlussfolgerung:

- Bei einem 3D-Komplett-Scan, mit dem eine Messumgebung in 3D abgebildet werden kann, würde ein Datenvolumen von ca. 16 MB erzeugt werden. Diese Menge könnte auf keinen Fall zwischen gespeichert werden.
- Ein 3D-Single-Scan würde ca. ein Datenvolumen von 23 KB erzeugen. Plus die bereits fixen 61 KB würden ca. 83 KB verbraucht sein, bevor überhaupt eine Funktion zur 3D-Erfassung geschrieben sein würde. Deshalb wurde entschieden, dass maximal die Daten eines 3D-Single-Scans zwischen gespeichert werden sollen.

Beschreibung	Größe/Einheit
Max. Scanbereich Laserscanner	180 °
Max. Scanbereich Powercubeeinheit	90°
Minimale Winkelschrittgröße	1/8°
Größe eines 3D-Punktes. X,Y,Z mit Remission	4*4 = 16 Bytes
Gesamtpunkte pro 3D-Single-Scan	180*8 = 1440 Punkte
Datengröße pro 3D-Single-Scan	1440*16 = 23040 Bytes
Datengröße pro 3D-Komplett-Scan	90*8*23040 ≈ 16 MB

Tabelle 3.1.: Der Speicherverbrauch bei einem 3D-Laserscan.

Weitere Schnittstellen

Die Funktionsweise der SPI-, Can-Bus- und TCP/IP-Schnittstelle wird in dieser Arbeit nicht näher erläutert. Eine Beschreibung dieser Kommunikationsschnittstellen, sowie weitere Informationen über den C32, der Powercubeeinheit und des Laserscanners können unter den angegebenen Literaturhinweisen in Tabelle 3.2 nachgelesen werden.

Komponente	Link
Can-Bus (Controller Area Network)	[CAN in Automation(CIA), 2008]
SPI (Serial Peripheral Interface)	[Wikipedia.org, 2008c]
TCP/IP (Transmission Control Protocol/Internet Protocol)	[Wikipedia.org, 2008d]
Erstellen einer Anwendung auf dem C32	[Sick Ibeo GmbH, 2005]
Powercubeeinheit	Kommunikation: [Schunk GmbH, 2007a] Funktionen: [Schunk GmbH, 2007b] Technische Details: [Schunk GmbH, 2008]
LRS-2100	[Sick Ibeo GmbH, 2004a] und [Sick Ibeo GmbH, 2004b]
C32-Prozessor (TMS320C32-60)	[Texas Instrument, 2004]

Tabelle 3.2.: Literaturhinweise für die Komponenten der 3D-Einheit.

3.5. Anforderungen

Für die Entwicklung der 3D-Erfassungssteuerung wurden folgende funktionale und nicht-funktionale Anforderungen festgelegt:

3.5.1. Funktionale Anforderungen

1. **Entwicklung einer Ansteuerung der Powercubeeinheit:**

Die Ansteuerung erfolgt über Can-Bus vom C32 aus, wo die 3D-Erfassungssteuerung ausgeführt wird. Die Firma Schunk GmbH stellt für die Can-Bus-Kommunikation eine Telegrammbeschreibung zur Verfügung [s. Schunk GmbH, 2007a].

2. **Entwicklung einer 3D-Erfassung:**

Die 3D-Erfassung beinhaltet neben einer Synchronisation der Daten der Powercubeeinheit und des Laserscanners, eine Verarbeitung der synchronisierten Daten zu 3D-Daten.

3. **Entwicklung einer Anwenderschnittstelle der 3D-Einheit:**

Die USP-Schnittstelle ist um einen „3D-Service“ zu erweitern, der auf dem Service „Application“ basiert. Für diesen 3D-Service müssen die 3D-Kommandos erstellt werden. Weiter ist ein Protokoll für den Datenaustausch zwischen dem Anwender und der 3D-Erfassungssteuerung zu entwerfen.

4. **Erstellung einer internen Systemarchitektur:**

Folgende Aufgaben muss die interne Systemarchitektur erfüllen:

- Zeitnahe Bedienung der Anwenderschnittstelle.
- Diktierung unter welchen Bedingungen eine 3D-Erfassung vom Benutzer gestartet, gestoppt und konfiguriert werden kann.
- Diktierung unter welchen Bedingungen die 3D-Erfassung ausgeführt bzw. die ermittelten 3D-Daten zum Anwender gesendet werden.

5. **Entwicklung einer Test-Anwendung:**

Um einen Test der 3D-Erfassungssteuerung ausführen zu können, ist eine Testanwendung zu entwickeln.

3.5.2. Nicht funktionale Anforderungen

Entwicklersicht

- Der Aufbau der 3D-Einheit sollte leichtverständlich für andere Entwickler sein.
- Der Systemaufbau soll leicht erweiterbar bzw. anpassbar für andere 3D-Systeme mit anderer Hardware sein.

Winkelauflösung und Echtzeit

- Die 3D-Daten sollen möglichst in Echtzeit ermittelt werden.
- Die Winkelauflösung eines 3D-Lasersans sollte sich am maximal möglichen des Laserscanners orientieren, die mit $1/8^\circ$ spezifiziert ist.

Anwenderschnittstelle

- Leichte Bedienbarkeit der Anwenderschnittstelle.
- Mögliche Fehleingaben vom Benutzer müssen abgefangen und leichtverständlich erklärt werden.
- Zustandsüberwachung für den Anwender muss möglich sein, um angemessen reagieren zu können.
- Die Tolerierung von Fehlern muss geprüft werden. Nicht jeder Fehler muss zum kompletten Ausfall der 3D-Einheit führen.
- Wissen über den internen Aufbau der 3D-Einheit soll für den Benutzer nicht von Belang sein.

3.6. Randbedingungen für die Entwicklung

Test- und Entwicklungsumgebung

Der für die 3D-Einheit vorgesehene LRS-2100 stand für die Bachelorarbeit nicht zur Verfügung. Statt einem LRS-2100 wurde ein LD-OEM-1000 verwendet. Bis auf die Lasersensorik, die auf dem LRS Outdoor-tauglich ist und eine höhere Reichweite erzielen kann, unterscheiden sich beide Systeme nicht.

Für einen Test standen neben dem Arbeitsplatz noch die Innenräume des Flurs der Firma Ibeo zur Verfügung.

Für die Entwicklung der Erfassungssteuerung auf dem C32 stand das Programm Code-Composer Version 4.10.37 zur Verfügung. Dies ist die von Texas-Instrument mitgelieferte Entwicklungsumgebung mit Debuggerfunktion für den C32-Prozessor.

Keine Änderungen an den Basefunction und der Firmware

Die Entwicklung der 3D-Einheit sollte sich ausschließlich auf dem C32 beschränken. Eingriffe in die Firmware oder in den Basefunction sollten nur im Sonderfall vorgenommen werden.

3.7. Zusammenfassung

Neben der Vorstellung des Laserscanners und der Powercubeeinheit, wurde gezeigt, welche Bewegungsgrade durch den Aufbau der 3D-Einheit zur Verfügung stehen, um einen 3D-Scan erzeugen zu können.

Die 3D-Erfassungssteuerung findet auf dem C32, dem einen von zwei Prozessoren des Laserscanners, statt. Die 3D-Erfassungssteuerung kommuniziert mit dem Anwender, der Powercubeeinheit und der Lasersensorik über den Messrechner, dem anderen Prozessor des Laserscanners. Die hardware- und softwaretechnischen Voraussetzungen für diese Kommunikation über den Messrechner bzw. einer Entwicklung auf dem C32 wurden dargestellt.

Weiter wurde beschrieben welche Anforderungen für eine Entwicklung einer 3D-Erfassungssteuerung bestanden und welche Randbedingungen für die Entwicklung und dem Test der 3D-Einheit vorhanden waren.

4. Design und Realisierung

In diesem Kapitel wird das Design und die Realisierung der 3D-Erfassung beschrieben. Zunächst erfolgt eine Übersicht über die 3D-Erfassung. Am Anfang wird der Ablauf der 3D-Erfassung abstrakt beschrieben und die Module der 3D-Erfassungssteuerung kurz dargestellt. Anschließend erfolgt eine detaillierte Beschreibung des Designs und der Realisierung der einzelnen Module.

4.1. Übersicht über die 3D-Erfassung

4.1.1. Modularer Aufbau

Für das Design und die Realisierung wurden die einzelnen Aufgaben der 3D-Erfassungssteuerung in Module zerlegt. Die modulare Zerlegung wurde aus folgenden Gründen gewählt:

- Unabhängige Entwicklung und separater Test der Module.
- Wiederverwendbarkeit der Module.
- Änderungen und Erweiterungen sind einfacher durchführbar.

4.1.2. Der Ablauf und die Module der 3D-Erfassung

Im Folgenden wird abstrakt der Ablauf der 3D-Erfassung beschrieben. Während dieser Darstellung werden die einzelnen Module der 3D-Erfassung vorgestellt und ihre Aufgaben erläutert. Abb. 4.1 und Abb. 4.2 stellen den abstrakten Ablauf der 3D-Erfassung grafisch dar.

1. 3D-Anwenderschnittstelle:

Über die Anwenderschnittstelle sendet der Benutzer 3D-Kommandos, um eine 3D-Erfassung zu konfigurieren, zu starten, zu stoppen und 3D-Daten entgegen zu nehmen.

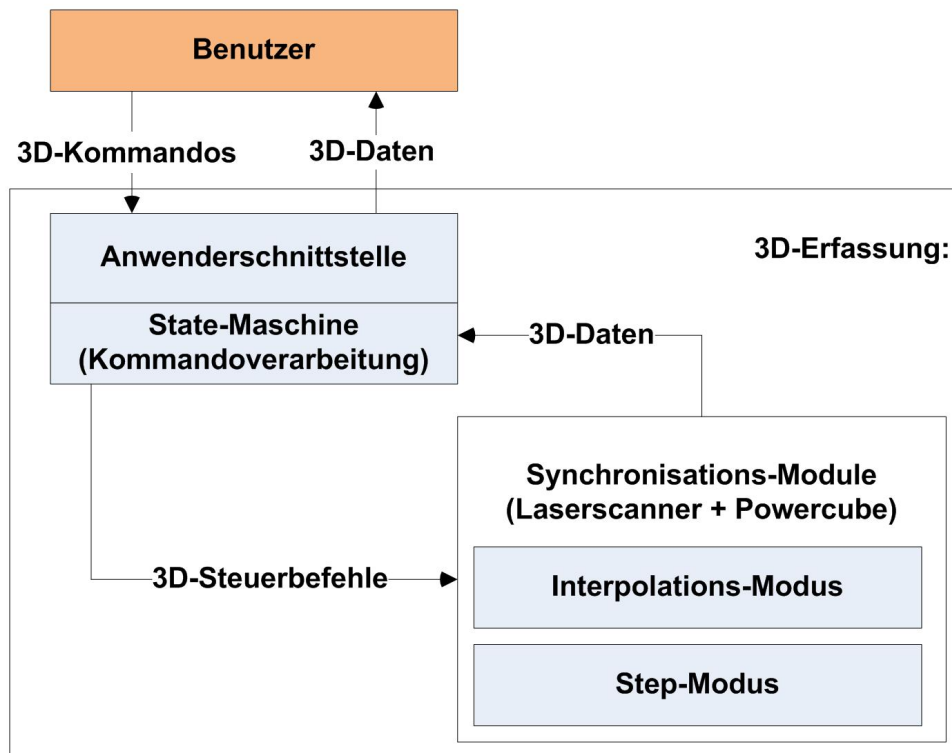


Abbildung 4.1.: Hauptmodule der 3D-Erfassungssteuerung.

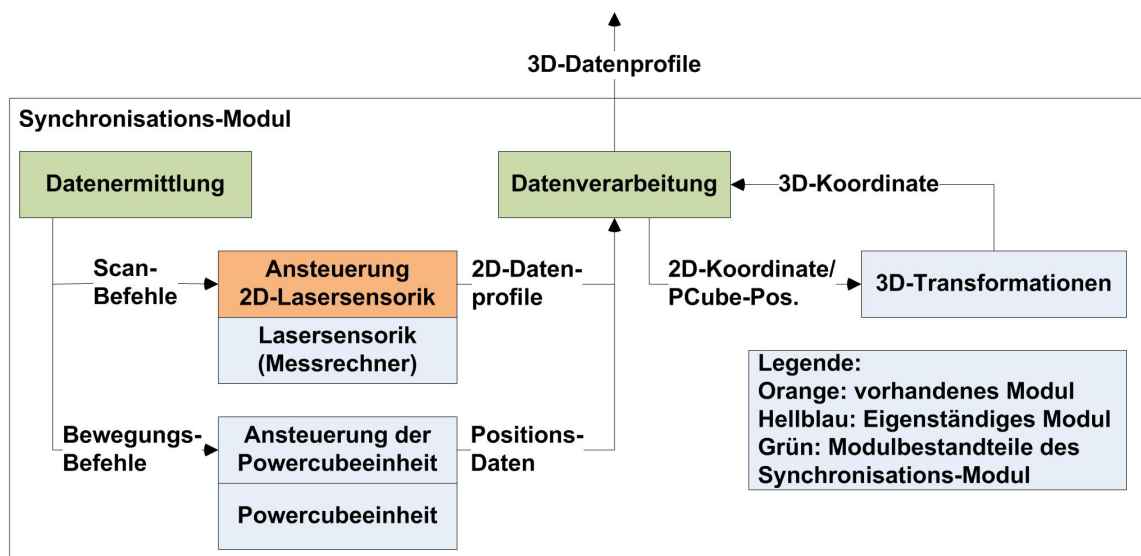


Abbildung 4.2.: Das 3D-Synchronisations-Modul.

2. State-Machine:

Die 3D-Kommandos werden zur State-Machine weitergeleitet. Diese sorgt für eine Verarbeitung der Kommandos und einen geregelten Ablauf der 3D-Erfassung.

3. 3D-Synchronisation:

Die eigentliche 3D-Erfassung wird im Synchronisations-Modul durchgeführt und koordiniert. Ein Synchronisations-Modul besteht intern, wie in Abb. 4.2 zu sehen ist, aus einer Datenermittlung und Datenverarbeitung. In der Datenermittlung wird die zeitliche Synchronisation des Laserscanners und der Powercubeeinheit durchgeführt. In jedem 3D-Single-Scan wird jeweils ein 2D-Datenprofil des Laserscanners erfasst und zeitgleich werden die Positionsdaten der Drehachse der Powercubeeinheit ermittelt. Diese beiden Datenstämme werden in der Datenverarbeitung dann zu einem 3D-Datenprofil verarbeitet. Aufgrund dessen, dass nur ein 3D-Datenprofil in der Erfassungssteuerung zwischen gespeichert werden kann (vgl. Punkt 3.4.2), wird das 3D-Datenprofil anschließend direkt zum Anwender gesendet. Das bedeutet, der Anwender muss die 3D-Datenprofile selber zu einem „Gesamt-3D-Profil“ zusammen fassen, um eine Messumgebung abbilden zu können. Um überhaupt eine 3D-Datenprofilerstellung zu ermöglichen, wurde eine Powercube-ansteuerung entwickelt und in einem separaten Modul ausgelagert. Die Ansteuerung der Lasersensorik war durch den Messrechner schon gegeben. Wie in Abb. 4.1 zu sehen, sind für die Synchronisation zwei verschiedene Module bzw. Methoden vorgesehen, um die 2D-Daten des Laserscanners und die Positionsdaten der Powercubeeinheit zu erfassen und zu synchronisieren.

Step-Modus: Im Step-Modus führt die Powercubeeinheit Schrittbewegungen entlang ihrer Drehachse durch. Nach jedem Schritt erfolgt ein Stopp der Drehachsenbewegung und der Laserscanner erfasst seine 2D-Daten. Diese 2D-Daten werden mit der aktuellen Position der Drehachse der Powercubeeinheit zu 3D-Daten verarbeitet.

Interpolations-Modus: Im Interpolations-Modus werden 3D-Daten erfasst bzw. synchronisiert während sich beide Einheiten gleichzeitig bewegen. Die Synchronisation erfolgt dabei mit Hilfe der linearen Interpolation.

4. 3D-Transformation:

Das 3D-Transformations-Modul ist ein Untermodul der 3D-Synchronisation und ist zuständig für die Verarbeitung einzelner 3D-Koordinaten. Nachdem die Daten des Laserscanners und der Powercubeeinheit synchronisiert wurden, wird das Modul „3D-Transformation“ genutzt, um aus den Daten beider Einheiten eine einheitliche 3D-Koordinate zu erstellen.

5. Test-Anwendung:

Um die 3D-Erfassung zu testen wurde eine PC-Testanwendung entwickelt, die später im Kap. 5.1 beschrieben wird.

4.1.3. Design und Realisierungsreihenfolge der Module

Die Reihenfolge des Entwurfs und der Realisierung der Module stand unter drei Gesichtspunkten:

- Realisierung eines Prototyps, der möglichst schnell einen Test der Zusammenführung der Module ermöglicht.
- Verfeinerung bzw. Optimierung des Prototyps.

Die Module wurden in folgender Reihenfolge entwickelt:

1. Anwenderschnittstelle und State-Machine:

Entwurf und Realisierung des Anwenderprotokolls mit zugehörigen 3D-Kommandos. Die Entwicklung eines Zustandsverhaltens auf die 3D-Kommandos. Diese beiden Module wurden parallel entworfen und verarbeitet, da sie eng miteinander verflochten sind.

2. Ansteuerung der Powercubeeinheit:

Test von Bewegungsbefehlen und Realisierung einer Can-Bus-Ansteuerung.

3. Step-Modus:

Entwicklung einer Synchronisationsart.

4. 3D-Transformationen:

Entwicklung der Basistransformationen und Transformation in das lokale Koordinatensystem der 3D-Einheit.

5. Entwicklung einer Testanwendung:

Test der Gesamtarchitektur. Die Testanwendung wird erst später im Kap. 5 Qualitätssicherung vorgestellt.

6. Interpolations-Modus:

Der Interpolations-Modus wurde nach hinten priorisiert, da mit dem Step-Modus bereits ein Modus zum Testen der Gesamtarchitektur bereit stand. Der Interpolations-Modus wurde aus Zeitgründen nur entworfen und nicht mehr realisiert.

4.2. 3D-Anwenderschnittstelle

Die 3D-Anwenderschnittstelle stellt für den Benutzer eine abstrakte Sicht auf die 3D-Einheit zur Verfügung. Wie die 3D-Einheit einen 3D-Laserscan erzeugt, spielt für den Benutzer keine Rolle. Beim Aufbau der 3D-Kommandos war daher das primäre Ziel, die Anwenderschnittstelle so allgemein wie möglich zu halten.

Die Kommunikation mit dem Anwender findet, wie bereits erwähnt, über TCP/IP statt und unterliegt den Richtlinien des User-Service-Protokolls. Entsprechend des User-Service-Protokolls wurde für die Anwenderkommunikation mit der 3D-Einheit ein neuer 3D-Service erstellt. Im ersten Abschnitt wird das neue 3D-Service-Protokoll erläutert und anschließend werden die erstellten 3D-Kommandos beschrieben.

4.2.1. Der Aufbau des 3D-Service-Protokolls

Eine 3D-Kommandosequenz besteht aus einer 3D-Request und aus einem 3D-Acknowledge. Eine 3D-Request ist ein 3D-Kommando, welches Daten der 3D-Anwendung parametrisiert bzw. anfordert. Eine Request wird immer mit einem Acknowledge beantwortet, es sei denn, die allgemeinen Richtlinien des User-Service-Protocols werden nicht eingehalten (z.B. die Checksumme ist falsch).

Die Größe eines 3D-Datenpakets bzw. der Datenteil einer TCP/IP-Datennachricht ist auf dem System auf maximal 10240 Bytes spezifiziert. Wie in Tabelle 3.1 bereits dargestellt wurde, könnten im maximalen aber mehr als 10240 Bytes Scan-Daten erzeugt werden. Daher kann es passieren, dass ein Acknowledge in zwei oder mehr Pakete aufgeteilt werden muss.

In Abb. 4.3 ist der Protokollaufbau von einer 3D-Request und von einem 3D-Acknowledge zu sehen. Die 3D-Kommando-ID entspricht einer Service-ID des User-Service-Protocols und ist 2 Bytes groß. Das erste Byte steht für die Identifikation des neuen 3D-Services und wurde mit 0x14 festgelegt. Dieser Wert entspricht einem möglichen Wert, der vom Service „Application“ vorgegeben wird [vgl. Kap.4 Sick Ibeo GmbH, 2006]. Das zweite Byte dient als Identifikation für das 3D-Kommando. Bei einem Acknowledge muss die Kommando-ID mit dem Operator OR mit einem Wert von 0x8000h verknüpft werden. Das wird vom User-Service-Protokoll vorgegeben. Ansonsten würde das Kommando vom Messrechner nicht zum Anwender gesendet werden.

Das Segment Errorcode existiert nur bei einem 3D-Acknowledge. Der Errorcode ist der Return-Code einer 3D-Request und sagt aus, ob das 3D-Kommando erfolgreich

Paketaufbau eines 3D-Request:

3D-Command-ID (2 Bytes)		Data len (2 Bytes)		Data 0 bis 10236 Bytes
Service-Code	Command-ID	High-Byte	Low-Byte	

Paketaufbau eines 3D-Acknowledge:

3D-Command-ID (2 Bytes)		Errorcode (2 Bytes)	Packet-ID (1 Byte)	Data len (2 Bytes)	Data 0 bis 10233 Bytes
Service-Code	Command-ID			High-Byte	Low-Byte

Abbildung 4.3.: Protokollaufbau von einem 3D-Request und einem 3D-Acknowledge.

verarbeitet wurde bzw. welcher Fehler auftrat. Für eine Beschreibung der Errorcodes (s. Anhang B). Die Paket-ID wird ebenfalls nur bei einem 3D-Acknowledge verwendet und dient zur Paket-Identifizierung, wenn ein 3D-Acknowledge in mehrere Pakete aufgeteilt werden muss. Eine ID von 0 entspricht dem letzten Paket einer 3D-Acknowledgepaketreihenfolge. Wird z.B. ein Acknowledge in zwei Pakete aufgeteilt, so nimmt die Paket-ID des ersten Paketes den Wert 1 und das zweite bzw. letzte Paket den Wert 0 an. Die Parameter „Data len“ und „Data“ sind selbsterklärend.

4.2.2. Der Aufbau eines 3D-Datenprofils

Wie bereits beschrieben, wird am Ende eines 3D-Single-Scans ein 3D-Datenprofil erstellt und zum Anwender gesendet. Das Datenprofil teilt sich in einen immer gleich großen Header und in einen variabel großen 3D-Scandatenbereich auf (s. Abb. 4.4).

Header

Der erste Parameter des Headers fungiert als 3D-Single-Scan-Index, während der zweite Parameter die Anzahl der 3D-Single-Scans in einem 3D-Laserscan darstellt. Angenommen für eine 3D-Datenerfassung einer Messumgebung werden zehn 3D-Single-Scans durchgeführt. Der dritte 3D-Single-Scan hätte den Wert 3, während der zweite Parameter des 3D-Datenprofil-Headers den Wert 10 haben würde. Der

3D-Single-Scan-Index	Data pro 3D-Single-Scan
Max. Number of 3D-Single-Scans	
Timestamp-Scanstart	
Timestamp-Scanend	
Number of 3D-Points	
X_mm oder R_mm	Data pro 3D-Point
Y_mm oder Φ in $^{\circ} * 1000$ (Winkel Laserscanner)	
Z_mm oder Θ in $^{\circ} * 1000$ (Winkel Powercube)	
Echo(Remission)	

Abbildung 4.4.: Paketaufbau von einem 3D-Datenprofil.

Anwender erhält durch diese beiden Parameter Auskunft darüber, wie viel Prozent einer Erfassung abgeschlossen ist. In einer wiederholenden Erfassung kann mit diesen beiden Parametern erkannt werden, ob ein Bereich erneut erfasst wurde. Die beiden nächsten Parameter des Headers bestehen aus zwei Zeitstempeln. Je zu Beginn und Ende des 3D-Single-Scans. Dadurch lassen sich die 3D-Datenprofile in einen zeitlichen Zusammenhang darstellen. Der letzte Parameter im Header gibt an, wie viele 3D-Messpunkte im variablen 3D-Scandatenbereich vorliegen.

3D-Scan-Daten

Der variable 3D-Scandateil besteht aus mehreren 3D-Messpunkten. Die Anzahl der 3D-Messpunkte ist variabel. Welche Daten ein 3D-Messpunkt enthält, das kann der Anwender mit der Auswahl aus 2 verschiedenen Optionen selber entscheiden.

Bei der Wahl der Option „X-Y-Z-Punkt“ wird, wie in Tab. 4.1 zu sehen ist, ein 3D-Messpunkt durch eine X-Y-Z-Koordinate und einem Echowert ausgedrückt. Die X-Y-Z-Koordinate ist dabei eine Koordinate, die sich im lokalen Koordinatensystem der 3D-Einheit befindet. Das lokale Koordinatensystem wird noch später ausführlicher im Modul „Transformation“ unter Punkt 4.7 beschrieben. Wird die zweite Option „Roh-Format“ gewählt, wird keine Transformation in das lokale Koordinatensystem der 3D-Einheit durchgeführt und die Daten liegen im polaren „Roh-Format“ der beiden Einheiten vor. Der Anwender muss dann selber eine Transformation in ein lokales Koordinatensystem durchführen.

Bytes	Option X-Y-Z	Option Roh-Format
Byte 0..3	X-Koordinate in mm.	Abstandskoordinate in $^{\circ} \cdot 1000$.
Byte 4..7	Y-Koordinate in mm.	Winkelkoordinate Laserscanner in $^{\circ} \cdot 1000$.
Byte 8..11	Z-Koordinate in mm.	Winkelkoordinate Powercubeeinheit in $^{\circ} \cdot 1000$.
Byte 12..15	Echowert.	Echowert.

Tabelle 4.1.: Darstellung eines 3D-Messpunktes unter den verschiedenen Optionen.

4.2.3. 3D-Kommandos

Die Beschreibung der 3D-Kommandos erfolgt in drei Abschnitten. Im ersten Abschnitt werden die allgemeinen Spezifikationen zu den 3D-Kommandos erläutert, die für jedes Kommando bindend sind. Im zweiten Abschnitt werden die realisierten 3D-Kommandos beschrieben, indem ihre Aufgaben erläutert werden. Der dritte Abschnitt befindet sich im Anhang D und fasst tabellarisch alle realisierten 3D-Kommandos zusammen.

Allgemeine Spezifikationen

4 Bytedatengröße pro Datenparameter:

Für die Größe eines Datenparameters eines 3D-Kommandos wurden 4 Bytes festgelegt. Der Grund dafür ist das nicht IEEE entsprechende 4-Byte-Datenformat des C32 (s. Kap. 3.4.2). Zwar werden nicht bei allen Parametern 4 Bytes benötigt und Speicherplatz wird innerhalb eines Datenpakets verschwendet, jedoch können die

Kommandos ohne viel Aufwand sofort in der 3D-Erfassungssteuerung weiterverarbeitet werden. Aufgrund dessen wird bei jedem Parameter die Eigenschaft Größe/Datentyp angegeben. Unter Größe wird angegeben wie viel Bytes für diesen Parameter verwendet werden und der Datentyp sagt aus, wie viel Speicherplatz tatsächlich genutzt wird.

Beispiel: 4 Bytes/Short

Der Parameter ist 4 Bytes groß. Short hat normalerweise eine Größe von 2 Bytes. Das bedeutet, nur die ersten 2 Bytes von 4 Bytes werden genutzt.

Erfolglöse Kommandos:

Ein nicht erfolgreiches Kommando enthält im Errorcodesegment einen Wert > 0 (0=Errorsuccess). Tritt ein Fehler auf, wird als Antwort eine „Get-State“ Antwort gesendet (vgl. Kommando Get-State unter Anhang D), die den Status der 3D-Einheit und eine Beschreibung des aufgetretenen Fehlers enthält.

Die erstellten 3D-Kommandos

Statusabfrage:

Der Status der 3D-Einheit besteht aus dem Working-Status, einem globalen Fehlerstatus und einem lokalen Fehlerstatus und dient neben der Statusabfrage auch zur Fehlerdiagnose.

Der Working-Status sagt aus, in welchem Zustand sich die 3D-Einheit befindet, z.B. ob gerade gemessen wird. Die Status werden in Abhängigkeit der Ausführungen der 3D-Kommandos erreicht und werden noch später in Kap. 4.3.3 ausführlich beschrieben. Der globale und lokale Fehlerstatus wird gesetzt, wenn ein Fehler in der 3D-Einheit aufgetreten ist. Der globale Fehlerstatus informiert über Fehler, die innerhalb der 3D-Erfassungssteuerung auftraten. Sie sind abhängig vom jeweiligen Anwenderkommando bzw. von der internen Ausführung. Der lokale Fehlerstatus informiert dagegen über einen Fehler, der in einer der angebotenen Komponenten (Powercubeeinheit oder Messrechner) auftrat. Sie entsprechen daher den jeweiligen Fehlerbeschreibungen der einzelnen Komponenten und dienen vor allem der Fehlerdiagnose. Alle drei Status werden mit dem 3D-Kommando „Get-State“ abgefragt (vgl. Anhang D).

Setzen und Lesen der 3D-Konfigurationsdaten:

Mit dem Kommando zum Setzen von 3D-Konfigurationen wird dem Anwender die Möglichkeit geboten, die 3D-Erfassung auf seine Messumgebung anzupassen. Um

die 3D-Erfassung auf sein gewünschtes Sichtfeld anzupassen, kann er die Scanbereiche der 3D-Einheit einschränken. Die Bereiche werden dabei der Einfachheit halber mit relativen Werten eingegrenzt. Für den Laserscanner kann z.B. ein Bereich von 0° bis 180° angegeben werden. Tatsächlich konfiguriert der Anwender jedoch indirekt einen Bereich zwischen 90° und 270° , der aufgrund der Sichtbegrenzung des Laserscanners möglich ist (vgl. Punkt 3.2). Die Auflösung einer Erfassung einer 3D-Messumgebung bestimmt der Anwender durch die Angabe der Winkelschrittweite. Als Auswahl wird dem Anwender eine Auflösung von 1° , $1/2^\circ$, $1/4^\circ$ und $1/8^\circ$ angeboten. Diese Auswahl orientiert sich an der Möglichkeit des Laserscanners und ist auch für die Bewegung der Powercubeinheit zutreffend. Für das Lesen und Schreiben von Konfigurationen wurde jeweils ein Kommando entworfen.

Zustandswechsel und das Abholen von 3D-Daten

Die Zustände der 3D-Erfassungssteuerung werden später in Kap. 4.3.3 ausführlich erläutert. Für einen Zustandswechsel wurden zwei Kommandos erstellt. Ein Kommando wurde für einen Wechsel in den Zustand „Idle“ und ein weiteres Kommando wurde für den Wechsel in den Zustand „3D-Measure“ entworfen. Weiter wurde für ein Abholen von den 3D-Datenprofilen, die im Step-Modus erfasst wurden, ein Kommando entworfen. Der Aufbau des 3D-Datenprofils wurde bereits in Kap. 4.2.2 beschrieben.

4.3. State-Machine

Die State-Machine bildet die interne Systemarchitektur der 3D-Erfassungssteuerung. Sie bildet das Bindeglied zwischen der Anwenderschnittstelle und der 3D-Erfassung (Synchronisation und Datenverarbeitung). Sie erfüllt die unter Punkt 3.5.1 beschriebenen Anforderungen der zeitnahen Bedienung der Anwenderschnittstelle und der Diktierung, unter welchen Bedingungen eine 3D-Erfassung vom Benutzer gestartet, gestoppt und konfiguriert werden kann. Das bedeutet, sie gibt an unter welchen Bedingungen ein 3D-Kommando verarbeitet wird. Für die Kommandoverarbeitung wurde das Automaten-schema nach Harel [s. Harel, 1998] für die State-Machine gewählt.

4.3.1. Wahl des Harelautomaten

Der Harelautomat wurde gewählt, weil mit ihm komplexes Zustandsverhalten einfach moduliert und dargestellt werden kann. Ein weiterer Hauptaspekt bei der Wahl war, dass das Automaten-schema verschiedene Reaktionsmöglichkeiten auf verschiedene Ereignisse bietet.

Auf ein Ereignis kann unterschiedlich reagiert werden, indem ...

- Nebenbedingungen gesetzt werden können.
- die Zustände das bisherige Geschehen speichern können.
- Randbedingungen gesetzt werden können, die zum Zeitpunkt des Auftretens eines Ereignisses herrschen sollen.

[vgl. S.2-4 Fohl, 2007].

Durch diese Eigenschaften können, wie später noch genauer beschrieben wird, durch Setzen von Bedingungen, 3D-Kommandos unterschiedlich verarbeitet und auf ein Fehlverhalten der 3D-Erfassung angemessen reagiert werden.

4.3.2. Ausdrücke des Harelautomaten und die Verwendung in der State-Machine

Transition, Aktionen und Aktivitäten

Eine Transition ist eine Teilreaktion auf ein erwartetes Ereignis und überführt im Regelfall den Automaten in einen neuen Zustand [vgl. S.6 Fohl, 2007]. Eine Transition in der 3D-Erfassungssteuerung ist ein Kommando, das einen Zustandswechsel hervorruft.

Eine Aktion und eine Aktivität sind zeitbehaftete Vorgänge im System. Die Aktionen sind an Transitionen gebunden, während Aktivitäten an Zustände gebunden sind [vgl. S.7 und S.8 Fohl, 2007]. In der State-Machine sind Aktivitäten und Aktionen Funktionen, die in Abhängigkeit vom Zustand und dem eingetroffenen Kommando ausgeführt werden. Das bedeutet, je nach Zustand ist es möglich, dass ein Kommando anders verarbeitet wird.

Entry, Exit, Do und throughout

Die Aktivitäten bzw. die Funktionen, die je nach Zustand ausgeführt werden, können durch Schlüsselwörter noch genauer beschrieben werden. Die Aktivitäten, die hinter den Schlüsselwörtern entry und exit stehen, werden ausgeführt, wenn ein Zustand betreten oder verlassen wird. Aktivitäten hinter den Schlüsselwörtern do und throughout werden ausgeführt, wenn sich im Zustand befunden wird. Die do-Aktivitäten sind zeitlich begrenzt und enden irgendwann, während throughout-Aktivitäten solange ausgeführt werden bis der Zustand verlassen wird [vgl. S.8 Fohl, 2007].

λ -Übergang und Wächter

Ein λ -Übergang ist ein spontaner Übergang. Ist ein Zustand mit einem λ -Übergang deklariert, so wird ein Zustandswechsel ausgeführt, wenn die do-Aktivität beendet ist. Ein Wächter ist eine Bedingung, die zu dem Zeitpunkt der Ausführung einer Aktivität oder einer Aktion erfüllt sein muss [vgl. S.12 und S.13 Fohl, 2007]. In der 3D-Erfassungssteuerung werden λ -Übergänge und Wächter kombiniert für die Fehlerbehandlung eingesetzt. Auftretende Fehler werden an Wächterbedingungen gebunden. An Hand der Wächterbedingung wird entschieden, wie schwerwiegend der Fehler ist und gegebenenfalls wird in den Zustand „System-Error“ gewechselt, der die 3D-Erfassungssteuerung außer Betrieb setzt. Dies ermöglicht eine bedingte Fehlertolerierung bzw. eine Fehlerkontrolle der 3D-Einheit.

4.3.3. Zustände des Automaten der 3D-Erfassungssteuerung

Der Harel-Automat der 3D-Erfassungssteuerung besteht, wie die Abb. 4.5 zeigt, aus folgenden Zuständen:

- **Start (Z0):**

Primäre Aufgabe des Startzustandes ist die Initialisierung der State-Machine, der Erfassungssteuerung, der Powercubeinheit und der Lasersensorik. Nach erfolgreichem Abschluss der Initialisierung wird automatisch in den Zustand IDLE gewechselt.

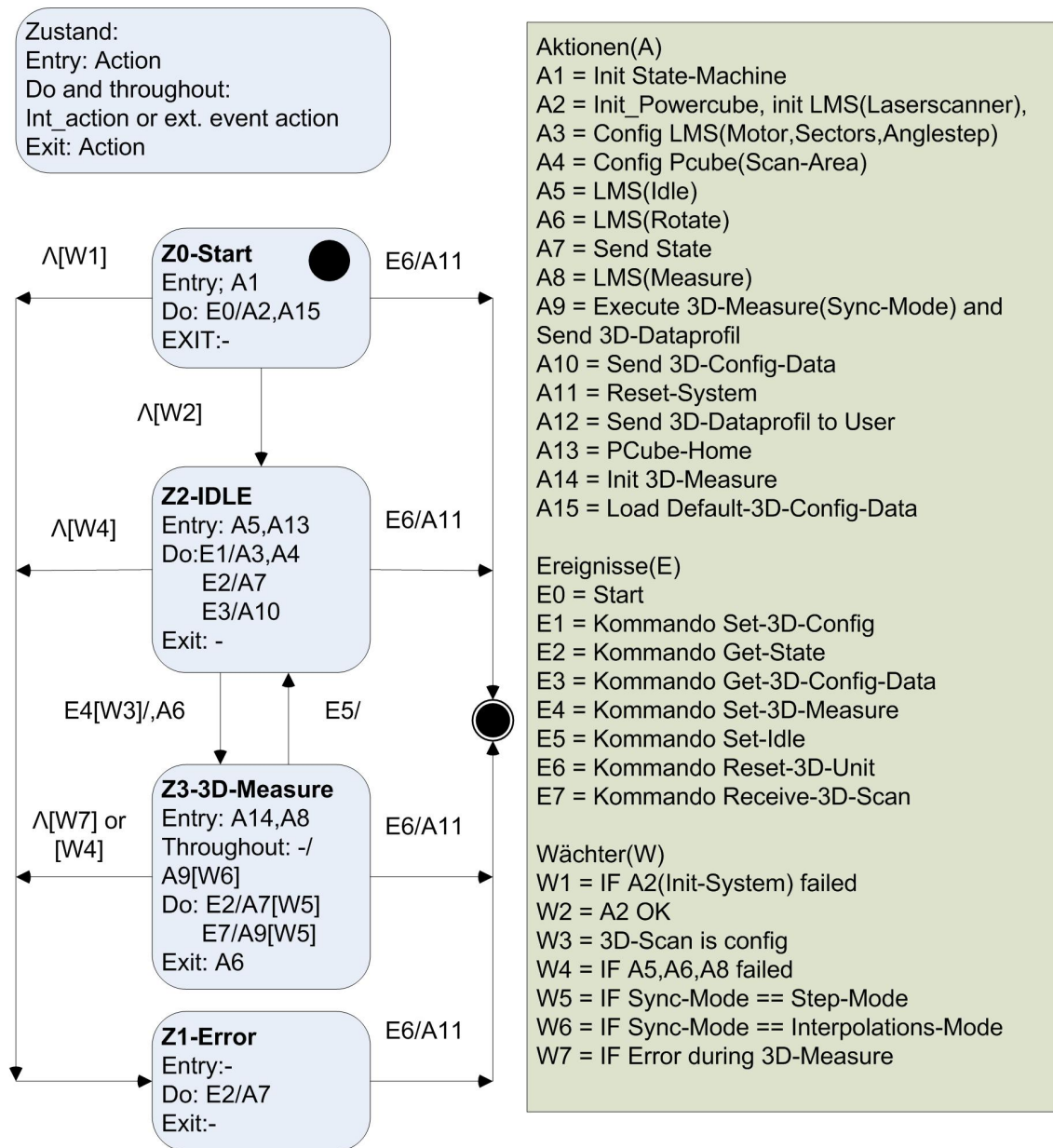


Abbildung 4.5.: Der Harelautomat der 3D-Erfassungssteuerung.

- **IDLE (Z2):**

Der IDLE-Zustand dient als Stromspar- und Konfigurationszustand. Bei Eintritt in den Zustand wird die Drehachse der Powercubeeinheit in die Referenzposition gefahren. Anschließend wird die Powercubeeinheit und der Laserscanner in den „Ruhezustand“ versetzt. Das Setzen und Lesen von Konfigurationen

für einen 3D-Laserscan kann nur in diesem Zustand durchgeführt werden. Wurde ein 3D-Laserscan erfolgreich gesetzt, kann mit dem Kommando „Set-3D-Measure“ ein 3D-Laserscan gestartet werden, indem die State-Machine in den Zustand „3D-Measure“ überführt wird.

- **3D-Measure (Z3):**

Bei einem Eintritt in den Zustand wird eine Erfassung einer 3D-Messumgebung initialisiert. Der Zustand 3D-Measure hat je nach Erfassungsbzw. Synchronisationsmethode ein anderes Verhalten.

Step-Modus: Im Step-Modus wird ein 3D-Single-Scan nur durchgeführt, wenn das Kommando „Receive-3D-Scan“ empfangen wird. Die Begründung dafür ist, dass während der Realisierung auffiel, dass eine Anwenderkommunikation über TCP/IP und eine Powercubekommunikation über Can-Bus nicht gleichzeitig möglich war. Es kam vor, dass Anwenderkommandos nicht in der 3D-Erfassungssteuerung am C32 ankamen bzw. im Messrechner „verschluckt“ wurden. Nach Ende des 3D-Single-Scans wird das 3D-Profil zum Anwender gesendet und somit das Kommando beantwortet.

Interpolations-Modus: Bei der Erfassung im Interpolations-Modus ist geplant, dass das 3D-Datenprofil gleich nach der Erfassung ohne Aufforderung zum Anwender gesendet wird. Während der Erfassung werden alle Kommandos, bis auf das Set-IDLE-Kommando (Abbruch der 3D-Erfassung) ignoriert. Grund dafür ist, das noch später ausführlicher beschriebene Zeitverhalten der Erfassung im Interpolations-Modus. Wie später noch zu sehen, ist geplant, dass alle 200 ms ein 3D-Datenprofil erzeugt wird. Die Zeitspanne für ein Warten auf ein Anwenderkommando, welches ein 3D-Datenprofil abholt oder die Beantwortung eines anderen Kommandos, wäre zu gering. Der zeitliche Ablauf wäre gestört.

- **System-Error (Z1):**

Befindet sich die Erfassungssteuerung in diesem Zustand, ist ein gravierender Fehler vorgefallen, z.B. ist eine Kommunikation zur Powercubeeinheit fehlgeschlagen. Eine Erfassungssteuerung kann nur mit einem Reset-Kommando zurückgesetzt werden. Ist ein Fehler aufgetreten und die 3D-Einheit ist nicht komplett ausgefallen, kann der Fehler mit dem Kommando „Get-State“ (vgl. Anhang D) ermittelt bzw. ausgelesen werden.

Interner Ablauf in der State-Machine

Wie in Abb. 4.6 zu sehen, führt die State-Machine immer wieder eine Schleife aus. Am Anfang der Schleife wird geprüft, ob ein Anwenderkommando (externes Ereignis) eingetroffen ist. Ist dies der Fall, wird die Funktion des aktuellen Zustandes aufgerufen, die für die Beantwortung von Kommandos zuständig ist. Ist kein Kommando vorhanden, wird die „interne Aktion“ des aktuellen Zustandes aufgerufen. Eine interne Aktion kann in der State-Machine nur ein 3D-Single-Scan sein. Wird in der externen bzw. internen Aktion eine Nachricht für den Anwender erstellt, so wird diese anschließend zum Anwender gesendet.

Durch diesen Ablauf wird gewährleistet, dass während einer 3D-Erfassung geprüft werden kann, ob ein 3D-Kommando vom Benutzer eingetroffen ist. Ansonsten, wenn kein 3D-Kommando zur Erfassungssteuerung gesendet wird, wird automatisch mit der 3D-Erfassung fortgefahren. Weiter bedeutet dieser Ablauf aber auch, dass eine Beantwortung eines Kommandos zeitlich von der Dauer einer internen bzw. externen Aktion abhängig ist. Ein 3D-Kommando wird erst entgegen genommen, wenn die aktuelle Aktion beendet wurde und eine Antwort wird erst gesendet, wenn eine Aktion beendet wurde.

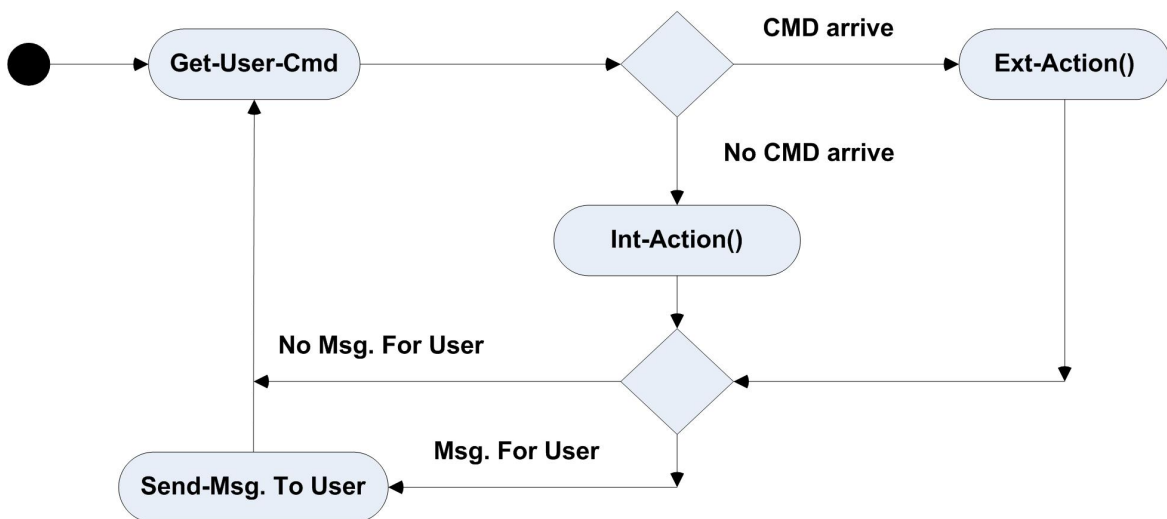


Abbildung 4.6.: Interner Ablauf der State-Machine der 3D-Erfassungssteuerung.

4.4. Ansteuerung der Powercubeeinheit

Die Ansteuerung der Powercubeeinheit wurde nach Vorbild der Vorgabe der Schunk GmbH [s. Schunk GmbH, 2007a] entwickelt. Diese beschreibt, wie Kommandos für eine Can-Bus-Kommunikation mit der Powercubeeinheit aufgebaut werden müssen.

Identifizier	CommandID	ParameterID bzw. MotionID	Data 0 ... Data 7
--------------	------------------	---------------------------	-------------------

Abbildung 4.7.: Der Aufbau einer Can-Nachricht an die Powercubeeinheit [vgl. Schunk GmbH, 2007a].

Wie in Abb. 4.7 zu sehen ist, wird im ersten Byte eines Can-Bus-Kommando an die Powercubeeinheit die Command-ID versendet. Die Command-ID gibt die Kommandoart an, während im zweiten Byte der Nachricht mit Hilfe einer Motion- bzw. Parameter-ID mitgeteilt wird, was für ein Kommando der Kommandoart ausgeführt werden soll. Für jede Kommandoart wurde eine Templatefunktion entworfen und umgesetzt. Um ein Kommando für die Powercubeeinheit zu erstellen, wird der zugehörigen Templatefunktion als Parameter die Motion- bzw. Parameter-ID eines Kommandos, ein Pointer auf einem Datenbuffer und die Länge des Datenbuffers übergeben. Weiter wurden für das Kopieren von Daten in oder aus dem Datenbuffer extra Kopierfunktionen entwickelt. Diese waren nötig, da in der Erfassungssteuerung, wie bereits schön erwähnt, kein IEEE-Format unterstützt wird.

Folgendes Beispiel zeigt die Realisierung des Bewegungskommandos „I-STEP“:

```
int PCube_MoveStepInc( int i_pos,unsigned short ui_time_ms)
{
    unsigned char data[6];
    c32_Copy_Int_To_CharArray(data,i_pos);
    c32_Copy_Short_To_CharArray(&data[4],ui_time_ms);
    return PCube_SetMotionCmd(ISTEP_MODE,data,6);
}
```

Durch diese Realisation mit den Templatefunktionen können Funktionen in der Ansteuerung der Powercubeeinheit schnell und leicht hinzugefügt werden.

4.5. Step-Modus

4.5.1. Grundprinzip

Im Step-Modus werden entlang der Drehachse der Powercubeeinheit mehrere „Scanpositionen“ festgelegt. Schritt für Schritt bewegt die Powercubeeinheit den Laserscanner von der einen in die andere Position. Jeweils, wenn eine Position erreicht wird, wird die Bewegung der Drehachse von der Powercubeeinheit gestoppt und ein 3D-Single-Scan wird durchgeführt. Der Laserscanner ermittelt dazu über einen 2D-Scan ein 2D-Datenprofil. Jede 2D-Koordinate des 2D-Datenprofils wird anschließend mit der aktuellen Position der Powercubedrehachse zu einer 3D-Koordinate erweitert. Das Ergebnis ist ein 3D-Datenprofil. Nachdem an jeder Position ein 3D-Datenprofil erfasst wurde, wurde ein 3D-Komplett-Scan ausgeführt, dessen Daten eine Messumgebung in 3D abbilden können.

4.5.2. Konzept und Realisierung des Step-Modus

Start:

Beim Start des Step-Modus wird zuerst die Start-Position angefahren. Die Start-Position ist die untere Grenze des Scanbereiches der Powercubeeinheit. Wird z.B. ein Scanbereich von 0° bis 45° festgelegt, so liegt die Startposition bei 0°.

Messvorgang:

Um ein 3D-Datenprofil zu einem 3D-Single-Scan erfassen zu können, wurde, wie in Abb. 4.8 zu sehen, folgender Ablauf in der 3D-Erfassungssteuerung auf dem C32 entworfen und realisiert.

1. Am Anfang wird überprüft, ob die gewünschte Position der Drehachse der Powercubeeinheit erreicht wurde. Ist dies nicht der Fall, wird solange gewartet, bis diese erreicht wurde.

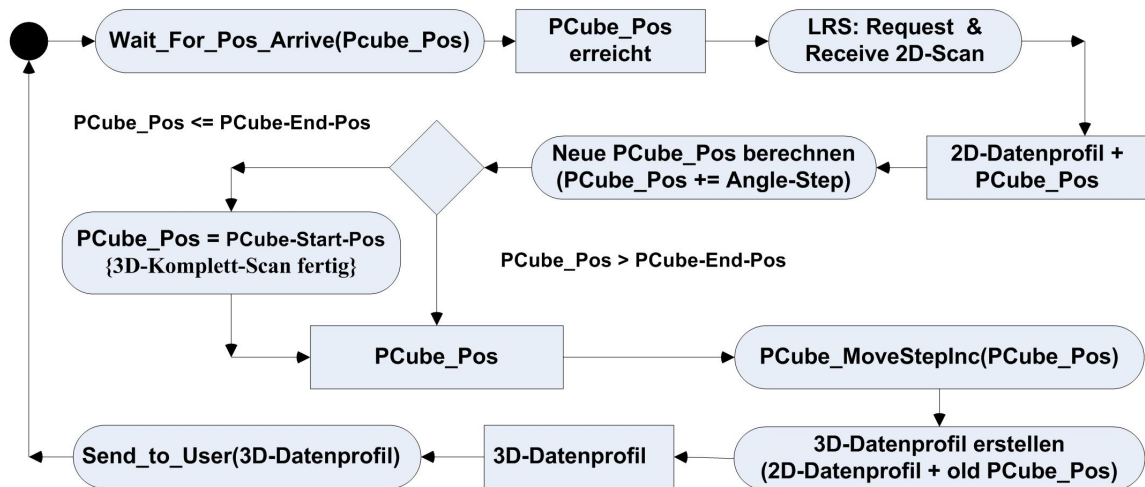


Abbildung 4.8.: Der 3D-Single-Scan im Step-Modus (Aktivitätsdiagramm UML 2.0).

2. Wie in Abb. 4.8 zu sehen, wird anschließend ein 2D-Scan durchgeführt. Die Drehgeschwindigkeit des Laserscanners ist dabei mit 10 Hz fest vorgeben und kann nicht von außen konfiguriert werden. Mit dem Kommando Request-Scan der Basefunction wird ein 2D-Datenprofil angefordert und mit dem Kommando Receive-Scan wird ein 2D-Datenprofil aus dem Buffer des Messrechners gelesen.
3. Anschließend wird die neue Position auf der Drehachse der Powercubeeinheit berechnet. Dazu wird die angegebene Winkelschrittweite zu der aktuellen Position addiert. Ist die errechnete Position größer als die Endposition, so ist die neue Position der Drehachse wieder die Startposition. In diesem Fall wäre eine 3D-Erfassung einer Messumgebung abgeschlossen und würde von neuen beginnen.
4. Nachdem die Position berechnet wurde, wird ein Fahrtbefehl mit der neuen Position zur Powercubeeinheit gesendet. Die Powercubeeinheit beginnt sofort mit der Fahrt in die gewünschte Position.
5. Während die Powercubeeinheit die nächste Position anfährt, wird parallel in der Erfassungssteuerung ein 3D-Datenprofil erstellt. Die 2D-Koordinaten des 2D-Laserscans werden dazu mit der zuvor aktuellen Position der Drehachse der Powercubeeinheit zu einem 3D-Datenprofil verarbeitet. Anschließend wird das erstellte Paket zum Anwender gesendet. Durch diese parallele Aktivität wird die Dauer eines 3D-Single-Scans deutlich verkürzt.

Powercubebewegungsart ISTEP

Als Bewegungsart der Powercubedrehachse wurde der vom Powercubekontroller bereitgestellte „ISTEP-Mode“ gewählt. Der „ISTEP-Mode“ bietet eine zeitgesteuerte Anfahrt einer Drehachsenposition. Als Parameter werden die Zielposition und die Zeit bis zum Eintreffen am Ziel angegeben. Der Vorteil dieser Bewegungsart ist, dass alle Schritte die gleiche Dauer haben und keine Geschwindigkeit und Beschleunigung gesetzt bzw. errechnet werden muss. Der Powercubekontroller berechnet selbständig die Geschwindigkeit und Beschleunigung, um in der vorgegebenen Zeit zum Ziel zu gelangen.

Nach einem Test mit verschiedenen Zeiten, wurde eine Dauer von 300 ms für eine Schrittbewegung festgelegt. Schnellere Bewegungen in kürzere Zeit führten zur einer hackenden bzw. einer fehlerhaften Schritt-Bewegung.

Die Zielposition im ISTEP-Mode wird in Inkrementen angegeben. Das Datenformat der Angabe ist dabei Integer. Der Benutzer gibt jedoch den Start- und Endwert für den Drehachsenbereich und der Winkelschrittweite in Grad an (vgl. Anhang D). Daher müssen die Positionswerte im Step-Modus von Grad in Inkremente umgerechnet werden.

45° entsprechen 201250 Inkremente. Daraus ergibt sich folgende Formel für eine Inkrement-Position.

$$\text{Inkrement_Pos} = (\text{Grad_Pos}/45) * 201250.$$

Die Berechnung wird im Float-Format durchgeführt und am Ende für die Übergabe in Integer gecastet. Dadurch ergibt sich eventuell eine Abweichung der Start-Position im Step-Modus. Bei der Umrechnung der Winkelschrittweite (1°, 1/2°, 1/4° oder 1/8°) kommt es ebenfalls zu einer Abweichung, die sich bei jeder Neuberechnung der Drehachsenposition addiert. Theoretisch ergibt sich dadurch eine maximale Abweichung von ca. 0,005° (s. Anhang C) in der Position der Powercubedrehachse für den letzten 3D-Single-Scan, wenn im maximalen Bewegungsbereich der Drehachse gescannt wird. Die Abweichung wurde aufgrund der geringen Größe als akzeptabel eingestuft.

4.6. Interpolations-Modus

Im Interpolations-Modus werden die Daten der Powercubeeinheit und des Laserscanners mit Hilfe der linearen Interpolation zu 3D-Daten synchronisiert, während sich der Laserscanner und die Powercubeeinheit gleichzeitig bewegen. Der parallele Ablauf des Messrechners, des Powercubekontrollers und der 3D-Erfassungssteuerung wird dabei gezielt genutzt, um einen 3D-Single-Scan in Echtzeit zu erfassen und zu verarbeiten. Aus Zeitgründen wurde der Interpolations-Modus nur entworfen, aber nicht mehr realisiert. Daher wird er nur theoretisch betrachtet.

4.6.1. Der 3D-Single-Scan

Die Aufgaben im Interpolations-Modus während eines 3D-Single-Scans sind folgendermaßen verteilt:

1. **Datenermittlung für die Interpolation:**

Der Messrechner und der Powercubekontroller werden so initialisiert, dass sie ohne weiteres Eingreifen der 3D-Erfassungssteuerung in einem festen Intervall Daten ermitteln, die mit Hilfe der linearen Interpolation zu 3D-Daten verarbeitet werden können.

2. **Datenverarbeitung der erfassten Daten:**

Parallel zur nächsten Ermittlung eines Datensatzes wird in der 3D-Erfassungssteuerung mit Hilfe der linearen Interpolation ein 3D-Datenprofil erstellt, welches anschließend zum Anwender gesendet wird.

Die Datenermittlung für die Interpolation

Um die Daten mit Hilfe der linearen Interpolation synchronisieren zu können, müssen die in Tabelle 4.2 beschriebenen Daten ermittelt werden.

Abkürzung	Bedeutung
2D-Data	entspricht einem 2D-Scan des Laserscanners mit mehreren 2D-Koordinaten, der in einem 2D-Datenprofil zusammengefasst ist.
TB	Zeitstempel zu Beginn des Laserscans(Bestandteil des 2D-Datenprofils).
TE	Zeitstempel zum Ende des Laserscans(Bestandteil des 2D-Datenprofils).
P0	Index des ersten 2D-Messpunktes bzw. der ersten 2D-Koordinate.
PEnd	Index des letzten 2D-Messpunktes bzw. der letzten 2D-Koordinate.
PC0	Powercubedrehachsenposition zum Beginn eines 2D-Laserscans.
PCE	Powercubedrehachsenposition zum Ende eines 2D-Laserscans.

Tabelle 4.2.: Die Daten, die für eine lineare Interpolation nötig sind.

Um diese Daten ermitteln zu können, werden folgende Funktionalitäten vom Laserscanner und der Powercubeeinheit zur Verfügung gestellt:

1. Die Powercubedrehachse kann in einer trapezförmigen Rampenbewegung bewegt werden (s. Abb. 4.9). Die Geschwindigkeit oberhalb der Rampe ist konstant. Die Phase oberhalb der Rampe wird in dieser Arbeit „Scan-Phase“ genannt.
2. Die Distanzen werden während eines 2D-Laserscans in festen zeitlichen Abständen erfasst. Ein Zeitstempel am Anfang und am Ende des 2D-Scans kann ebenfalls festgehalten werden[vgl. Kap.5.3.2 Sick Ibeo GmbH, 2006].
3. Die aktuelle Position der Drehachse der Powercubeeinheit kann jeweils bei einer fallenden und steigenden Flanke über eine Leitung getriggert werden. Das bedeutet, es können zwei Positionswerte der Drehachse zu unterschiedlichen Zeitpunkten erfasst werden und später abgeholt werden [vgl. Schunk GmbH, 2007b]. Die Triggerleitung der Powercubeeinheit ist mit dem Messrechner verbunden. So kann exakt zu einem Winkel, während eines 2D-Laserscansvorgangs, eine Position der Drehachse getriggert werden.

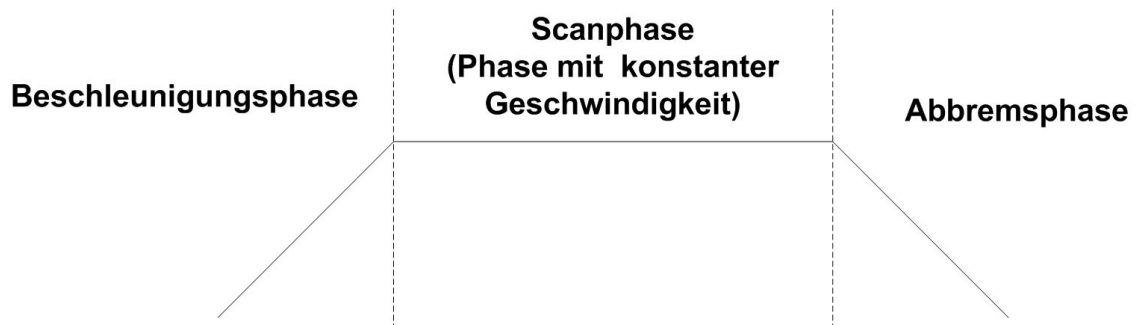


Abbildung 4.9.: Die Rampenbewegung der Powercubedrehachse.

Diese Funktionalitäten werden folgendermaßen vom Messrechner und der Powercubeeinheit kombiniert:

Wie in Kap. 3.2 beschrieben, kann bei der Erfassung eines 2D-Datenprofils nur in einem Bereich zwischen 90° bis 270° während der 360° Laserkopfumdrehung gemessen werden. Aufgrund dessen ist geplant, dass die Zeitstempel (TE und TB) und Positionsdaten der Drehachse der Powercubeeinheit (PC0 und PCE) wie in Abb. 4.10 zu sehen, folgendermaßen ermittelt werden:

Bei 90° in einem 2D-Scan wird jeweils...

- der Beginn-Zeitstempel (TB) des 2D-Laserscans erfasst.
- mit einer fallenden Flanke die Position der Powercubedrehachse getriggert (PC0) und im Controller der Powercubeeinheit gespeichert.

Bei 270° in einem 2D-Scan wird jeweils...

- der Ende-Zeitstempel (TB) des 2D-Laserscans erfasst.
- mit einer steigenden Flanke die Position der Powercubedrehachse getriggert (PCE) und im Controller der Powercubeeinheit gespeichert.

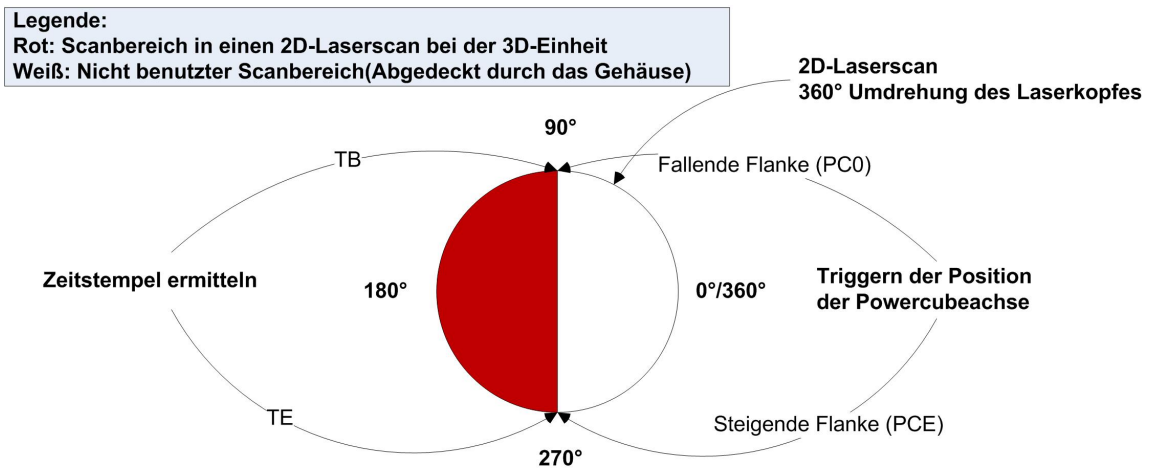


Abbildung 4.10.: Die Datenermittlung während eines 2D-Laserscans im Interpolations-Modus.

Gültig sind die ermittelten Daten aber nur, wenn sich während der Ermittlung die Rampenbewegung in der Scan-Phase befunden hat. In der Abbremsphase und in der Beschleunigungsphase ist die Positionsänderung der Drehachse aufgrund der unterschiedlichen Geschwindigkeit nicht mehr linear und eine Synchronisation auf Basis der linearen Interpolation kann nicht mehr durchgeführt werden. Um das zu überprüfen kann mit folgender Formel eine Sollposition für die zweite getriggerte Position (PCE) der Drehachse ermittelt werden:

$$\text{Soll_Pos} = (TE - TB) \cdot \text{Geschwindigkeit} [^\circ / \text{ms}] + PC0$$

Stimmt die Sollposition nicht innerhalb gewisser Grenzen mit der zweiten getriggerten Position überein, so befand sich die Rampenbewegung während der Datenermittlung entweder ganz oder teilweise in der Beschleunigungs- bzw. in der Abbremsphase.

Datenverarbeitung durch Interpolation der Drehachsenposition zu jeder 2D-Koordinate

Nachdem eine Datenermittlung abgeschlossen wurde und der Messrechner und die Powercubeinheit bereits einen neuen Datensatz für die Interpolation erfassen, wird parallel in der 3D-Erfassungssteuerung, wie in Abb. 4.11 zu sehen, folgender Ablauf durchgeführt, um die ermittelten Daten zu synchronisieren und zu einem 3D-Datenprofil zu verarbeiten.

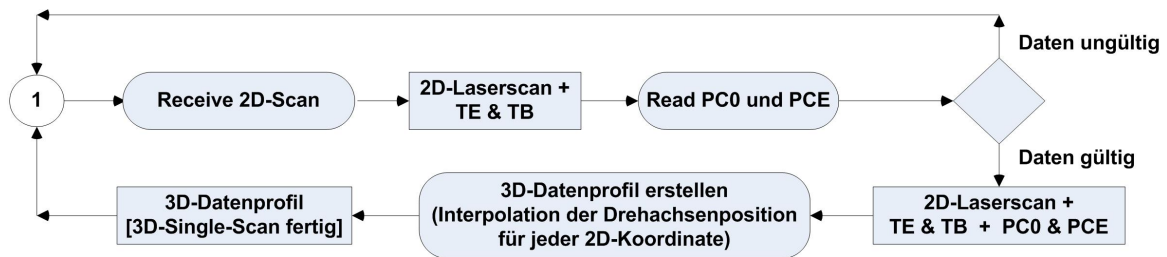


Abbildung 4.11.: Der Ablauf der Erfassungssteuerung auf dem C32 im Interpolations-Modus (Aktivitätsdiagramm UML 2.0).

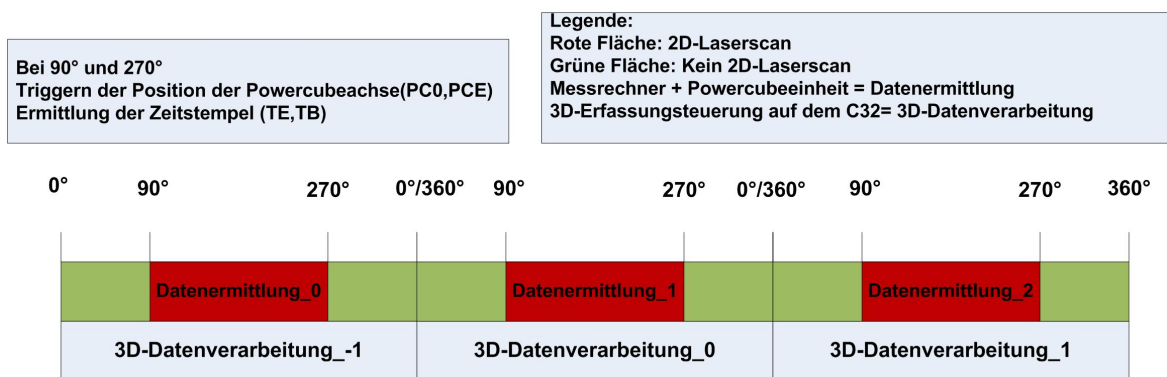


Abbildung 4.12.: Zeitliche Betrachtung der Datenermittlung und der Datenverarbeitung im Interpolations-Modus.

Der Synchronisierungspunkt zwischen Messrechner und Powercubeinheit und der Erfassungssteuerung ist die Funktion „Receive 2D-Scan“. Diese Funktion blockiert solange, bis ein 2D-Scan abgeschlossen ist. Das bedeutet ebenfalls, wie in Abb. 4.12 zu sehen ist, dass der Zyklus für die Verarbeitung der 3D-Daten in der Erfassungssteuerung abgeschlossen sein muss, bevor der Messrechner einen 2D-Scan beendet hat und neue Daten zur Verfügung stellt.

Die Funktion „Receive 2D-Scan“ kommt mit einem 2D-Datenprofil zurück. Anschließend werden die beiden Positionswerte der Powercubedrehachse aus der Powercubeinheit ausgelesen und nach der zuvor beschriebenen Methode auf Gültigkeit überprüft. Sind die Daten gültig, erfolgt die Erstellung des 3D-Datenprofils. Dazu wird zu jeder 2D-Koordinate des 2D-Laserscans die jeweilige Position der Drehachse der Powercubeinheit auf folgender Art und Weise linear interpoliert:

1. An Hand der beiden Zeitstempel (TE und TB) wird zu einer 2D-Koordinate, ausgedrückt durch Index (Px), der aktuelle Zeitstempel (T2D) durch folgende Gleichung linear interpoliert:

Endpunkte für die lineare Interpolation: (P0,TB) und (PEnd,TE).

$$T2D = TB + \frac{(TE - TB)}{(PEnd - P0)} \cdot (Px - P0)$$

2. Der Zeitstempel (T2D) zu der 2D-Koordinate wird im zweiten Schritt benutzt, um die aktuelle Powercubedrehachsenposition (PCX) zu der 2D-Koordinate zu erhalten. Wieder wird der gesuchte Wert mit Hilfe der linearen Interpolation ermittelt und folgende Gleichung wird dazu aufgestellt:

Endpunkte für die lineare Interpolation: (TB,PC0) und (TE,PCE).

$$PCX = PC0 + \frac{(PCE - PC0)}{(TE - TB)} \cdot (T2D - TB)$$

Folgendes Beispiel verdeutlicht, wie ein Positionswert der Drehachse zu einer 2D-Koordinate interpoliert werden kann.

Ein 2D-Scan eines 2D-Laserscanners hat folgende Eigenschaften:

- 181 Punkte (z.B. 180°-Scanbereich mit 1° Winkelauflösung) \implies PEnd = 180.
- Drehgeschwindigkeit: 5 Hz = 200ms \implies 100ms/180° \implies TE = 100ms, TB = 0ms.

Die Powercubeeinheit liefert folgende Werte:

- Position beim Start des 2D-Scans (PC0) = 0°.
- Position beim Ende des 2D-Scans (PCE) = 45°.

Gesucht wird die Position der Powercubedrehachse(PCX) bei 90° des 2D-Laserscans (Px = 90).

Die Position der Drehachse wird folgendermaßen berechnet:

$$T2D(90) = 0 + \frac{(100ms - 0ms)}{(180 - 0)} * (90 - 0) = 50ms$$

$$PCX = 0 + \frac{(90 - 0)}{(50ms - 0ms)} * (25ms - 0ms) = 22,5^\circ$$

4.6.2. Erfassung einer Messumgebung

Um eine Messumgebung möglichst in einer kurzen Dauer ohne große Lücken abtasten zu können, muss die Durchführung der 3D-Single-Scans koordiniert werden. Das Zeitraster wird dabei von der Drehgeschwindigkeit des Laserscanners vorgegeben. Unter der Annahme, dass die Drehgeschwindigkeit auf 5 Hz (minimale Geschwindigkeit) eingestellt wird, dauert die Datenerfassung für die Interpolation und die parallele Verarbeitung der Daten 200 ms.

Zu berücksichtigen ist, wie in Abb. 4.12 zu sehen ist, dass nur zwischen 90° und 270° aufgrund der Sichtbegrenzung des Laserscanners eine Lasermessung durchgeführt wird. Das bedeutet zeitlich betrachtet, dass zwischen 50 ms (90°) und 150 ms (270°) des 200 ms dauernden Datenermittlungsintervalls eine Lasermessung stattfindet. In den anderen 100 ms entsteht eine „Scanlücke“.

Weiter muss beachtet werden, dass die Beschleunigungsphase und Abbremsphase der Rampenbewegung möglichst gering gehalten wird. Dadurch wird die Dauer der Scan-Phase erhöht und mehr gültige 3D-Single-Scans können durchgeführt werden. Dies würde einer Erhöhung der Messauflösung entsprechen.

Unter der Voraussetzung, dass ein 2D-Laserscan und die Rampenbewegung exakt aufeinander abgestimmt werden können, kann eine Koordinierung mehrerer 3D-Single-Scans exemplarisch folgendermaßen aussehen:

Eigenschaften des 2D-Laserscanners:

- Drehgeschwindigkeit: 5 Hz.
- 2D-Laserscan misst in maximalen Scanbereich von 90° bis 270° die Distanzen.

Eigenschaften der Powercubeeinheit:

- Der Bewegungsbereich der Drehachse der Powercubeeinheit ist konfiguriert von 0° bis 40°.
- Die konstante Geschwindigkeit der Rampenfahrt ist auf 10° pro 200 ms eingestellt.
- Die Beschleunigung und die Abbremsung der Rampenfahrt wird so konfiguriert, dass jeweils 5° für die Beschleunigungsphase und der Abbremsphase gebraucht werden.

Aus diesen Eigenschaften resultiert, dass ein 3D-Single-Scan 200 ms dauert und während des 3D-Single-Scans eine Strecke von 10° entlang der Drehachse der Powercubeinheit zurück gelegt wird. Eine 3D-Messumgebung kann nun, wie in Abb. 4.13 zu sehen, durch eine Ausführung von zwei 3D-Komplett-Scans erfasst werden. Im ersten 3D-Komplett-Scan führt die Powercubeinheit über ihre Drehachse eine Schwenkung des Laserscanners von 0° bis 40° aus. Im zweiten 3D-Komplett-Scan erfolgt eine Schwenkung zurück zur 0° -Position. In jedem der zwei 3D-Komplett-Scans werden jeweils drei 3D-Single-Scans ausgeführt.

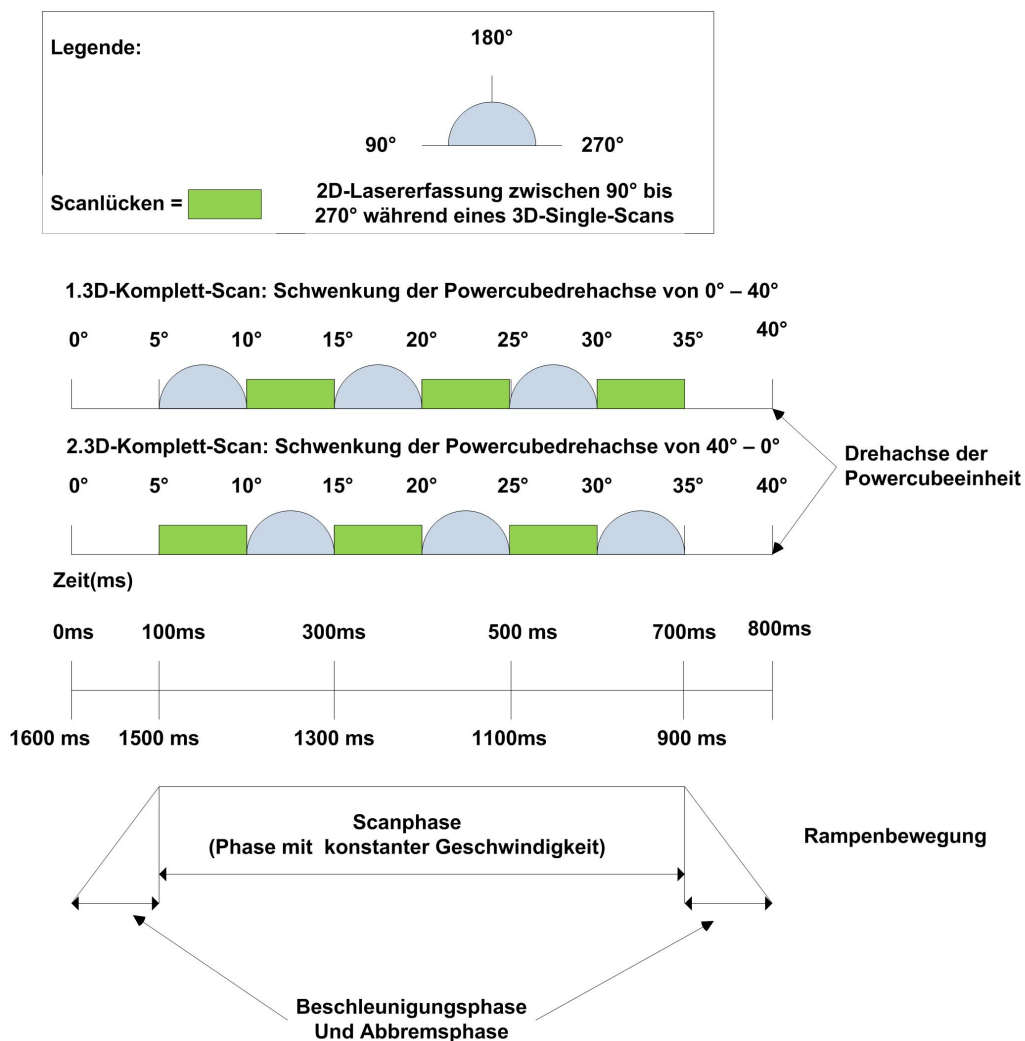


Abbildung 4.13.: Ein Beispiel für eine mögliche Koordinierung mehrerer 3D-Single-Scans im Interpolation-Modus.

Aufgrund der Sichtbegrenzung des Laserscanners wird innerhalb eines 3D-Single-Scans nur in einem 5°-Bereich der 10°-Strecke eine Lasermessung durchgeführt werden. In den anderen 5° entsteht eine Scanlücke. Aufgrund dieser Tatsache erfolgt während des ersten 3D-Komplett-Scans zwischen 5°-10°, 15°-20° und 25°-30° eine Erfassung der Messumgebung und 3D-Daten werden produziert. Zwischen 10°-15°, 20°-25° und 30°-35° finden keine Lasermessungen statt und Scanlücken entstehen.

Bei 700 ms beginnt die Rampenbewegung der Powercubedrehachse mit der Abbremsung, die bei 800 ms zu einem Stopp der Drehachsenbewegung führt. Anschließend wird sofort die Beschleunigung wieder gestartet. Dieser 200 ms dauernde Abbrems- und Beschleunigungsvorgang führt dazu, dass ein 3D-Single-Scan für ungültig erklärt wird und bei 35° ein neuer 2D-Laserscan eines 3D-Single-Scans statt findet. Durch diese Koordinierung werden auf dem Rückweg die Scanlücken des ersten 3D-Komplett-Scans geschlossen.

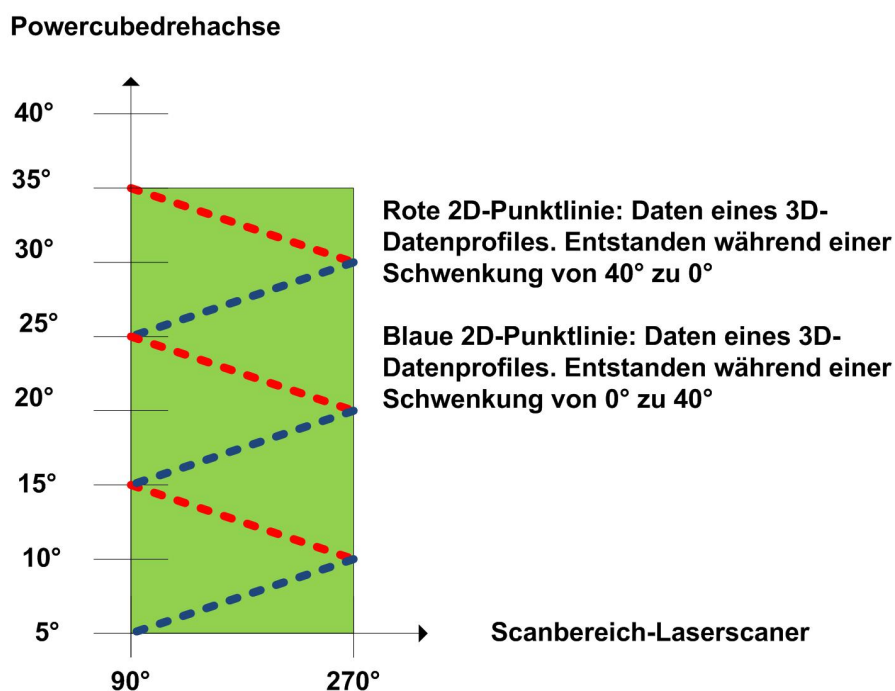


Abbildung 4.14.: Ergebnis der 3D-Erfassung des Beispiels für den Interpolationsmodus.

Am Ende des zweiten 3D-Komplett-Scans stehen mehrere 3D-Datenprofile zur Verfügung. Jedes Datenprofil würde, wie in Abb. 4.14 zu sehen ist, eine schräge 2D-Punktlinie darstellen.

4.6.3. Offene Punkte für die Implementierung

Bevor mit der Umsetzung des Entwurfs des Interpolations-Modus begonnen werden kann, müssen folgende Punkte gelöst werden:

1. Synchronisation der Rampenbewegung mit dem 2D-Laserscan:

Das Konzept der Koordinierung der 3D-Single-Scans für das Erfassen einer Messumgebung sieht vor, dass die Rampenbewegung der Powercubedrehachse und ein 2D-Laserscan aufeinander abgestimmt werden kann.

Dafür muss noch ein Konzept entworfen werden, wie dies am besten realisiert werden kann. Voraussetzung für ein Konzept ist eine Analyse der Rampenfahrt. Folgende Eigenschaften müssen ermittelt werden:

- Welche Geschwindigkeiten und welche Beschleunigungen können gefahren werden.
- Ermittlung der Dauer der verschiedenen Phasen unter verschiedenen Geschwindigkeiten und Beschleunigungen.
- Dauer von der Initialisierung einer Rampenfahrt bis zur Scanphase.

Aus diesen Daten muss die optimale Rampenbewegung ermittelt werden, die möglichst über eine kurze Beschleunigungs- und Abbremsphase verfügt. Im zweiten Schritt muss ein Synchronisationspunkt zwischen der Rampenfahrt und dem 2D-Laserscan gefunden werden. Das Startverhalten der Rampenfahrt muss dem Startverhalten der Laserscan-Durchführung angepasst werden, so dass z.B., wie im Beispiel der Erfassung einer Messumgebung, der Eintritt in die Scan-Phase der Rampenbewegung und die Lasermessung unter dem 90°-Winkel gleichzeitig stattfinden.

2. Realisierung der Powercubefunktionen für die Rampenbewegung:

Für die Konfiguration und Steuerung der Rampenbewegung müssen die jeweiligen Powercubefunktionen in der Erfassungssteuerung umgesetzt werden. Die bisherige Powercubeschnittstelle muss dafür erweitert werden. Die Rampenkommandos erwarten Parameter-Übergaben in IEEE-Float-Format. Dafür muss die mitgelieferte Assemblerfunktion von Texas Instruments zum Konvertieren von dem C32-Float-Format in das IEEE-Float-Format noch eingebunden werden.

4.7. 3D-Transformation

Das Modul „3D-Transformation“ ist für die Verarbeitung eines 3D-Messpunktes zuständig. Nach der Synchronisation des Positionswertes der Powercubedrehachse zu einer 2D-Koordinate des Laserscanners, liegen die Daten in dem Koordinatensystem ihrer jeweiligen Einheit vor. Im nächsten Schritt muss diese neue 3D-Koordinate in ein lokales Koordinatensystem der 3D-Einheit übertragen werden, um einen Zusammenhang zwischen der 2D-Koordinate und der Position der Powercubedrehachse herzustellen. Im Folgenden wird beschrieben, wie das lokale Koordinatensystem der 3D-Einheit in dieser Bachelorarbeit definiert wurde und wie eine neue 3D-Koordinate in dieses transformiert wird.

4.7.1. Definierung des lokalen Koordinatensystems der 3D-Einheit

Der Ursprung des lokalen Koordinatensystems ist ein lokaler Punkt der Einheit. Als lokaler Punkt und somit der Ursprung des lokalen Koordinatensystems wurde die rechte hintere und obere Kante des Gehäuses festgelegt gewählt (s. Abb. 4.15). Durch die Annahme, dass sich die 3D-Einheit bei einem 3D-Laserscan oberhalb der Messumgebung befindet, werden die drei Achsen folgendermaßen definiert:

- X-Achse ist die Breitenachse.
- Y-Achse ist die Höhenachse bzw. Längenachse.
- Z-Achse ist die Tiefenachse bzw. Distanzachse.

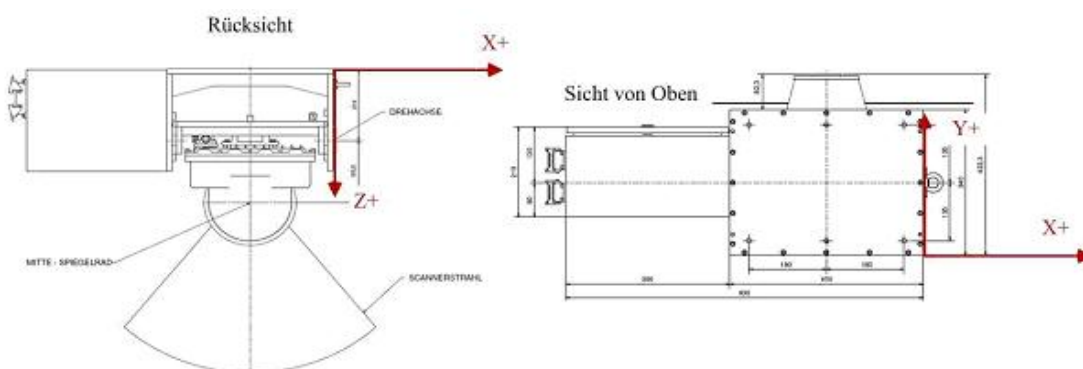


Abbildung 4.15.: Das lokale Koordinatensystem der 3D-Einheit.

4.7.2. Transformation in das lokale Koordinatensystem der 3D-Einheit

Um einen 3D-Punkt in das lokale Koordinatensystem zu übertragen, wurden vorab die folgenden Basistransformationen umgesetzt:

- 3D-Transformation einer 3D-Koordinate vom polaren in das kartesische Koordinatensystem.
- 3D-Rotation um die X-, Y- und Z-Achse.
- 3D-Translation entlang der X-,Y- und Z-Achse.

Die Transformation einer 2D-Koordinate des Laserscanners in eine 3D-Koordinate der 3D-Einheit besteht aus fünf Translationen und einer Rotation und insgesamt drei Schritten:

1. Transformation der polaren 2D-Koordinate des Laserscanners in eine kartesische 3D-Koordinate des Laserscanners:

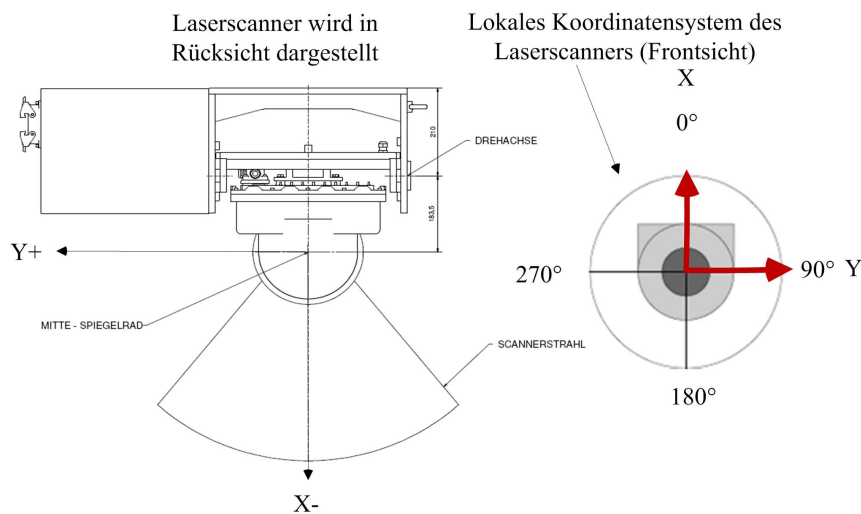


Abbildung 4.16.: Die Laserscannerausrichtung bei der 3D-Einheit.

Der Ursprung des lokalen Koordinatensystems des Laserscanners befindet sich, wie in Abb. 4.16 zu sehen ist, in der Mitte des Laserkopfes. Der erste Schritt ist die Transformation der polaren 2D-Koordinate in eine kartesische 3D-Koordinate. In der 3D-Perspektive befindet sich jede 2D-Koordinate des Laserscanners 90° zur Z-Achse. Im kartesischen Format bedeutet dies eine

Position von 0 auf der Z-Achse. Die Transformation wird durchgeführt, indem zuerst der polare 2D-Wert des Laserscanners um einen 90° -Winkel zu einer polaren 3D-Koordinate erweitert wird und anschließend in das kartesische Koordinatensystem übertragen wird.

2. Transformation zur Home-Position der Powercubeeinheit:

Bei einer Position von 0° befindet sich die Drehachse der Powercubeeinheit in der Home-Position. In der Home-Position befindet sich jede Achse des Koordinatensystems des Laserscanners parallel zu einer Achse des Koordinatensystems der 3D-Einheit. In diesem Fall kann die neue 3D-Koordinate des Laserscanners mit einer Translation in das lokale Koordinatensystem der 3D-Einheit übertragen werden. Da jedoch ein 2D-Laserscan im Winkel verschoben zur Home-Position der Powercubedrehachse aufgenommen werden kann, muss im nächsten Schritt die neue 3D-Koordinate zuerst in die Home-Position der Drehachse rotiert werden. Um die Rotation ausführen zu können, muss zuerst der Ursprung der 3D-Koordinate von der Mitte des Laserkopfes zur Mitte der Powercubedrehachse verschoben werden. Dazu werden zwei Translationen gleichzeitig ausgeführt. Die neue 3D-Koordinate wird, wie in Abb. 4.17 zu sehen ist, 185,2 mm entgegen der Z-Achse und 183,5 mm entlang der X-Achse verschoben. Anschließend erfolgt die Rotation der 3D-Koordinate um die Y-Achse in die Home-Position der Drehachse der Powercubeeinheit.

3. Transformation von der Home-Position in das lokale Koordinatensystem der 3D-Einheit

Im dritten Schritt erfolgt die Transformation der 3D-Koordinate in das lokale Koordinatensystem der 3D-Einheit. Dazu werden, wie in Abb. 4.18 zu sehen ist, gleichzeitig drei Translationen ausgeführt.

- Translation um 210 mm entlang der X-Achse auf die Höhe der Gehäusedecke.
- Translation um 225 mm entgegengesetzt der Y-Achse zur rechten Kante der Gehäusedecke.
- Translation um 170 mm entgegengesetzt der Z-Achse zur hinteren Kante der Gehäusedecke.

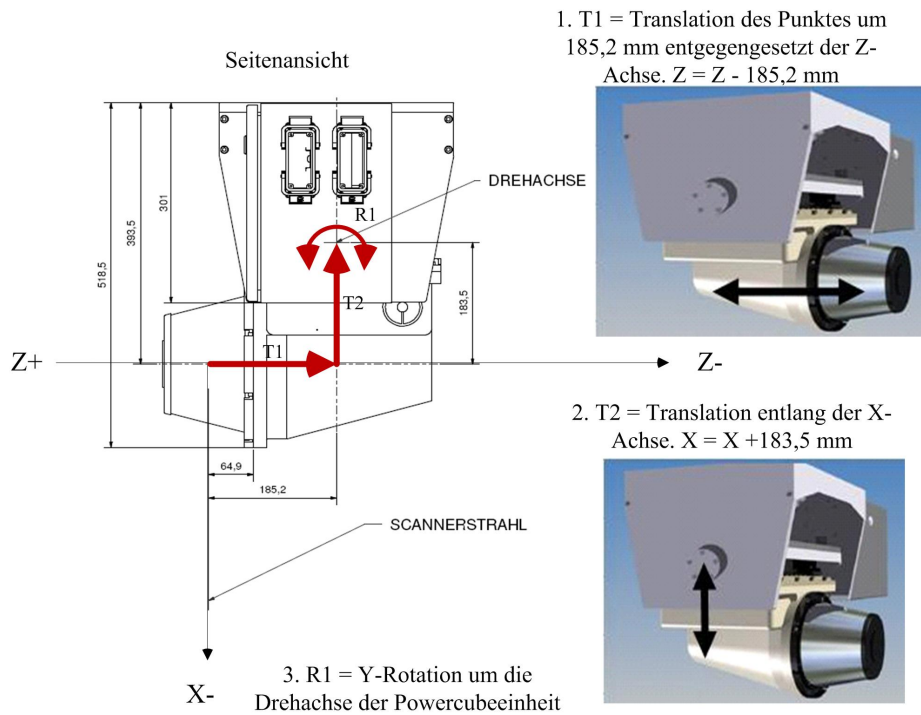


Abbildung 4.17.: Transformation in die Homeposition der Powercubedrehachse.

Nachdem die Translationen abgeschlossen sind, müssen die Achsen der 3D-Koordinate angepasst werden, damit sie mit den Achsen des Koordinatensystems der 3D-Einheit übereinstimmen. Dazu werden die Achsen folgendermaßen zugewiesen:

$$X' = -Y$$

$$Y' = Z$$

$$Z' = -X$$

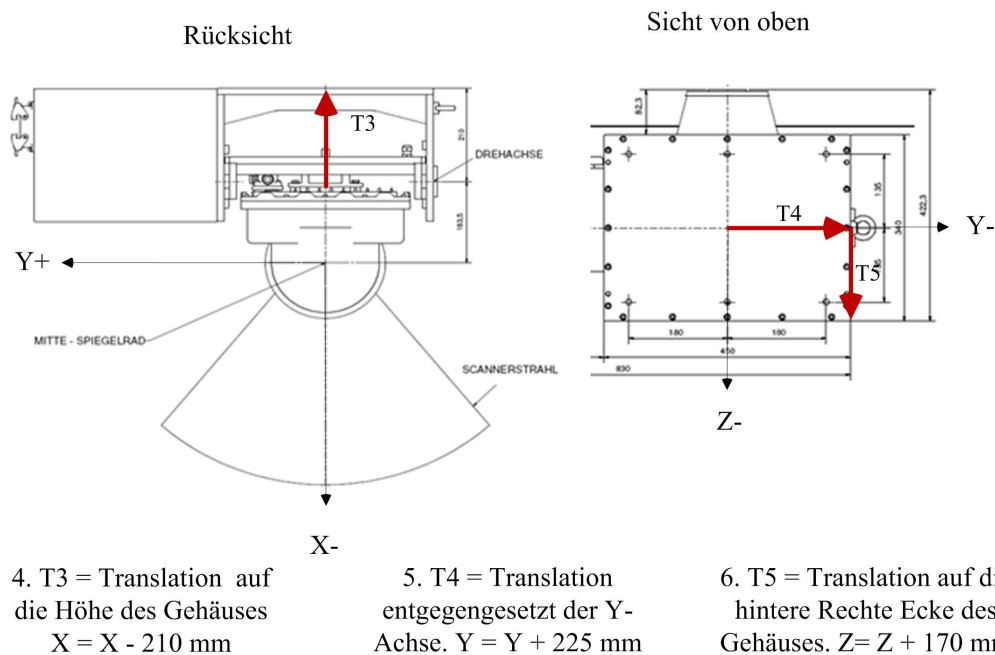


Abbildung 4.18.: Die Translation von der Home-Position zur rechten, hinteren Gehäusedecke.

4.8. Zusammenfassung

In diesem Kapitel wurde das Design und die Realisierung der 3D-Erfassung beschrieben. Die einzelnen Module der 3D-Erfassung wurden dargestellt. Im Modul 3D-Anwenderschnittstelle wurde erläutert, wie das Datenprotokoll zum Anwender aufgebaut ist und welche 3D-Kommandos dem Anwender für die 3D-Einheit zur Verfügung gestellt werden.

Weiter wurde gezeigt, wie die ankommenden 3D-Kommandos in der State-Machine unter welchem Zustand verarbeitet werden und wie ein geregelter Ablauf der 3D-Erfassung durchgeführt wird.

Wie die Daten der Powercubeeinheit und des Laserscanners ermittelt, synchronisiert und zu 3D-Daten verarbeitet werden, wurde in den Modulen Step-Modus und 3D-Transformationen dargestellt. Im Modul Interpolations-Modus wurde der Entwurf einer weiteren Synchronisationsmöglichkeit vorgestellt, um die Daten der Powercubeeinheit und des Laserscanners zu „verheiraten“.

5. Qualitätssicherung

In diesem Kapitel wird dargestellt, wie die 3D-Einheit getestet wurde. Zuerst erfolgt eine Beschreibung der Test-Anwendung, die für den Test der 3D-Einheit erstellt wurde. Anschließend werden die durchgeführten Tests erläutert und die Testergebnisse dargestellt.

5.1. Aufbau der Testanwendung

Die Testanwendung wurde in Visual C++ .Net entwickelt und besteht aus verschiedenen Modulen.

5.1.1. Testmodul Anwendersteuerung

Dieses Testmodul enthält eine Klasse, die eine Ansteuerung der 3D-Einheit bzw. eine Umsetzung der 3D-Kommandos beinhaltet. Um einen schnellen Test der 3D-Kommandos mit verschiedenen Parametern zu ermöglichen, wurden die Testfunktionen, wie in Abb. 5.1 zu sehen ist, an eine GUI (Grafic-User-Interface) gebunden.

Zudem wird die Kommunikation mit der 3D-Einheit geloggt. Dies ermöglicht eine Analyse der gesendeten und empfangenen Nachrichten.

5.1.2. Testmodule für einen 3D-Laserscan

Um die Synchronisation zwischen der Powercubeinheit und dem Laserscanner zu testen, wurden zwei Testmodule entwickelt. Das eine Modul dient zur 3D-Datenerfassung und ermöglicht eine Ausführung eines 3D-Laserscans. Die 3D-Datenprofile der einzelnen 3D-Single-Scans werden dabei entgegengenommen, gespeichert und nach der Annahme sofort visualisiert. Für die Visualisierung der 3D-Daten wurde das Tool Gnuplot [Gnuplot, 2008] Version 4.2.3 verwendet. Die

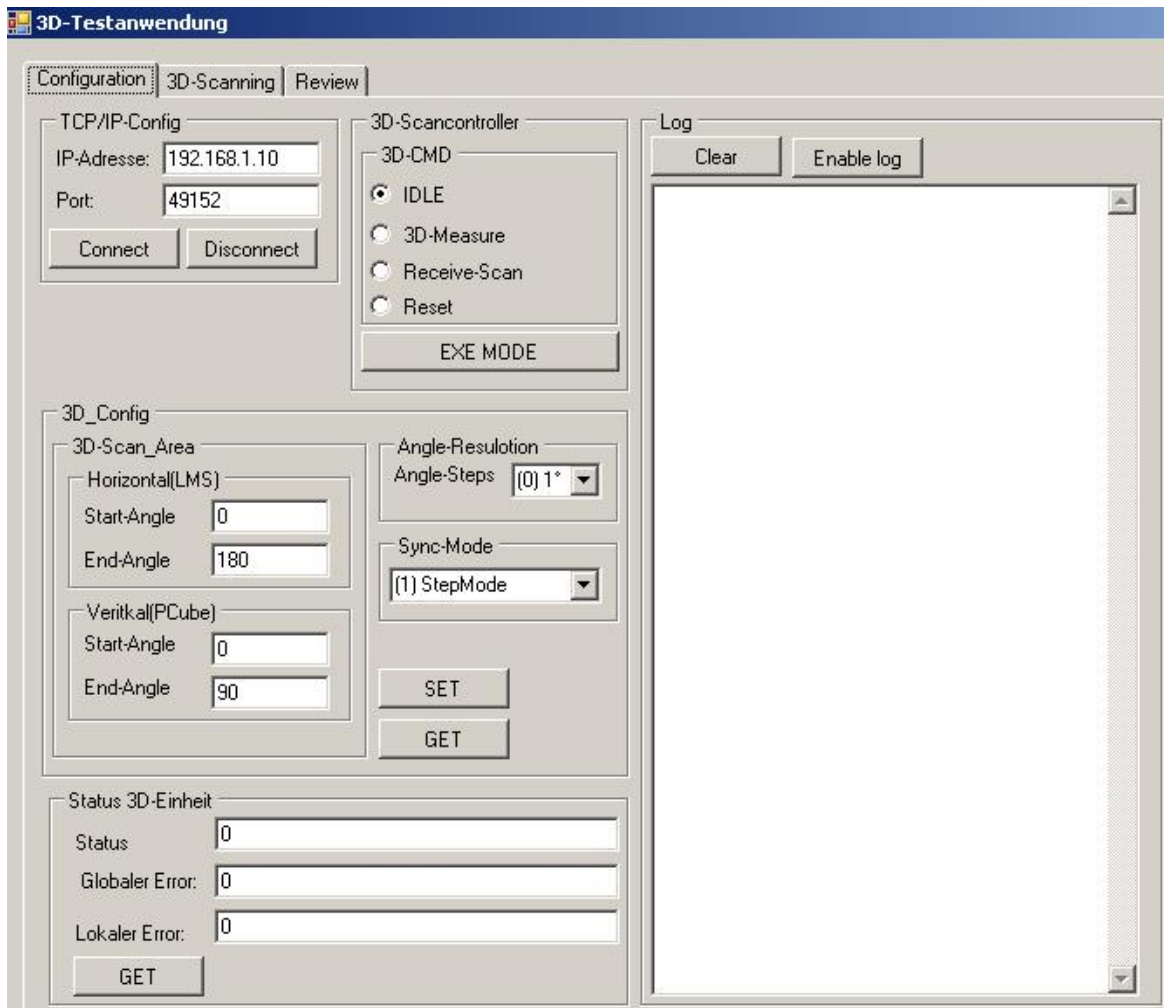


Abbildung 5.1.: Bild der GUI der Testanwendung

Visualisierung der 3D-Punkte der 3D-Datenprofile über Gnuplot ermöglicht eine sofortige Einschätzung, ob während der 3D-Erfassung größere Fehler aufgetreten sind.

Um ermittelte Daten später erneut auszuwerten, wurde ein „Review-Modul“ entwickelt. Einzelne 3D-Datenprofile können geladen und visuell verglichen werden.

5.2. Beschreibung der durchgeführten Tests

Das Testen der 3D-Einheit fand ausschließlich in Innenräumen der Firma Sick Ibeo statt. Ein Test unter Outdoorbedingungen wurde nicht durchgeführt. Während dieser Tests wurde die Fehlerkontrolle der State-Machine (vgl. 4.3.3) genutzt, um mögliche Fehler zu ermitteln. An Hand der Errorcodes, die mit dem 3D-Kommando „Get-State“ ausgelesen werden können, konnten Rückschlüsse gezogen werden, ob die 3D-Kommandos richtig verarbeitet wurden oder im Step-Modus Fehler auftraten.

5.2.1. Test der Anwenderschnittstelle und der State-Machine

Die Anwenderschnittstelle und die State-Machine wurden getestet, indem verschiedene Testfälle entworfen wurden, die fehlerhafte und richtige 3D-Kommandos simulieren. Bei jedem Testfall wurde ein 3D-Kommando ausgeführt und ein Sollergebnis festgelegt, welches mit dem Testergebnis verglichen wurde. Bei jedem Test wurde kontrolliert, ob das 3D-Kommando fehlerfrei empfangen, ausgewertet und gemäß seiner Spezifikation verarbeitet und beantwortet wurde. Die Testfälle lassen sich in drei Kategorien unterteilen:

- **Testkategorie 1: Test des 3D- Kommandos mit gültigen Parametern unter einem gültigen Zustand:**

Bei einem Test in dieser Testkategorie wird erwartet, dass das 3D-Kommando in der 3D-Erfassungssteuerung ordnungsgemäß verarbeitet wird und mit Errorcode „Error-Success“ beantwortet wird.

Beispiel: Setzen von Konfigurationen mit dem Kommando „Set-3D-Config“ im Status „Idle“: Das Kommando Set-3D-Config wird mit gültigen Parametern (Scanbereiche und Winkelauflösung sind innerhalb ihrer Spezifikation) zur 3D-Erfassungssteuerung gesendet. Die 3D-Erfassungssteuerung befindet sich im Status „Idle“. Nur in diesem Status ist eine Konfiguration zulässig und das Kommando wird verarbeitet. In diesem Fall wird erwartet, dass der Laserscanner gemäß übergebenen Parametern einen 2D-Scan konfiguriert und die 3D-Erfassungssteuerung für einen möglichen 3D-Laserscan vorbereitet wird. Die Erfassungssteuerung müsste das Kommando nach erfolgreicher Umsetzung mit einem Errorcode „Error-Success“ beantworten.

- **Testkategorie 2: Test des 3D-Kommandos mit gültigen Parametern unter einem ungültigen Zustand:**

Bei einem Test in dieser Testkategorie wird erwartet, dass das 3D-Kommando abgelehnt und mit einer Fehlermeldung quittiert wird.

Beispiel: Setzen von Konfigurationen mit dem Kommando „Set-3D-Config“ im Status „3D-Measure“: Während eines 3D-Laserscans ist kein Setzen von Konfigurationen erlaubt. Die Erfassungssteuerung müsste das Kommando ablehnen und ein Acknowledge mit einem Errorcode „ERR-3D-INVALID-COMMAND“ zum Anwender senden.

- **Testkategorie 3: Test des 3D- Kommando mit ungültigen Parametern im gültigen Zustand:**

Die Parameter eines 3D-Kommandos befinden sich bei einem Test in dieser Kategorie außerhalb ihrer Spezifikation. Die Erfassungssteuerung müsste das Kommando ablehnen und ein Acknowledge mit einem Errorcode „ERR-3D-INVALID-PARAM“ zum Anwender senden.

Beispiel: Setzen von ungültigen Konfigurationen mit dem Kommando „Set-3D-Config“ im Status „Idle“: Das Kommando wird mit ungültigen Set-3D-Config-Parametern zur 3D-Erfassungssteuerung gesendet. Z.B. könnte der Scanbereich für den Laserscanner mit einem falschen Wert angegeben sein. Die Erfassungssteuerung müsste erkennen, dass die Angabe des Laserscannerbereiches außerhalb der Spezifikation liegt, das Kommando ablehnen und eine Fehlermeldung zum Anwender senden.

5.2.2. Test der 3D-Erfassung im Step-Modus

Der Test des Step-Modus überprüft nur die Funktionalität des Step-Modus und die Transformation der ermittelten Daten in das lokale Koordinatensystem der 3D-Einheit. Die Überprüfung, ob Messobjekte erkannt werden, wurde nur visuell durchgeführt, indem mit der Testanwendung die 3D-Daten angezeigt wurden. Für den Test Step-Modus wurden verschiedene Testfälle entworfen, die den Step-Modus unter verschiedenen Bedingungen testet. Konfiguriert wurde der Step-Modus während dieser Testfälle über die getesteten Module der Anwenderschnittstelle und der State-Machine. Auf einen separaten Einzeltest mit fehlerhaften Übergabeparametern wurde verzichtet, da die Anwenderschnittstelle und die State-Machine fehlerhafte Angaben bereits filtern.

Ein Testfall ermöglicht einen Test folgender Funktionen:

- Synchronisation der Powercubeeinheit und des Laserscanners.
- Transformation der synchronisierten Daten in das lokale Koordinatensystem der 3D-Einheit.
- Erstellung und Versendung eines 3D-Datenprofils in einem 3D-Single-Scanvorgang.
- Erfassung einer Messumgebung, in der verschiedene Distanzen gemessen werden.

Allgemein beschrieben wurden folgende Konfigurationen getestet:

- Test mit verschiedenen Winkelauflösungen (1° , $1/2^\circ$, $1/4^\circ$, $1/8^\circ$).
- Laserscanner: Konfiguration verschiedener Scanbereiche.
- Powercubeeinheit: Konfiguration verschiedener Bewegungsbereiche.

Überprüfung der Funktionalitäten im Step-Modus

Die Synchronisation der Powercubeeinheit und des Laserscanners im Step-Modus wurde überprüft, indem kontrolliert wurde, ob der Laserscanner gemäß den Konfigurationen ein 2D-Datenprofil erfasst. Dazu wurden die Winkelauflösung und die Anzahl der Punkte mit einem definierten Sollergebnis verglichen. Weiter wurde überprüft, ob die jeweils nächste Position der Powercubedrehachse gemäß den Vorgaben berechnet, angefahren und einem 2D-Datenprofil zur Datenverarbeitung zugewiesen wird.

Die Transformation der synchronisierten Daten beider Einheiten in das lokale Koordinatensystem der 3D-Einheit wurde kontrolliert, indem stichprobenartig überprüft wurde, ob die Transformation ordnungsgemäß durchgeführt wurde. Dazu wurden die Translationen und die Rotation um die Drehachse der Powercubeeinheit bei jeder einzelnen Stichprobe nacheinander kontrolliert. Weiter wurde mit der visuellen Anzeige überprüft, ob die Werte der einzelnen Achsen eines 3D-Messpunktes prinzipiell zur Messumgebung passen.

Die Transformation der polaren 2D-Koordinate des Laserscanners in die kartesische 3D-Koordinate wurde dabei mit dem Tool „Calc 3D Pro“ [s. Greuer, 2007] überprüft. Weiter wurde kontrolliert, ob die 3D-Datenprofile in verschiedenen Größen erstellt und zum Anwender versendet werden können.

5.3. Zusammenfassung der Testergebnisse

Am Ende der Qualitätssicherung wurde folgender Status bei der Entwicklung der 3D-Einheit erreicht:

Anwenderschnittstelle und State-Machine

Die Anwenderschnittstelle und State-Machine funktionieren innerhalb ihrer Spezifikationen. Die 3D-Kommandos werden unter verschiedenen Status und verschiedenen Parametern, sowohl fehlerhaften und richtigen, in der 3D-Einheit entsprechend verarbeitet.

Transformation in das lokale Koordinatensystem der 3D-Einheit

Die Stichproben ergaben, dass die Transformation der synchronisierten Daten der Powercubeeinheit und des Laserscanners in das lokale Koordinatensystem der 3D-Einheit erfolgreich durchgeführt wird.

3D-Erfassung im Step-Modus

Der grundlegende Algorithmus für die 3D-Erfassung im Step-Modus funktioniert einwandfrei. Die Anforderung an die Winkelauflösung kann vom Algorithmus erfüllt werden. 3D-Profile können mit einer Winkelauflösung 1° , $1/2^\circ$, $1/4^\circ$, $1/8^\circ$ ermittelt werden.

Abb. 5.2 zeigt z.B. grafisch, wie im Step-Modus erfolgreich zwei 3D-Datenprofile ermittelt und übereinander gelegt wurden. Ein 3D-Datenprofil wird durch eine Punktlinie dargestellt und als Messumgebung für die Erfassung diente der Flur der Firma Sick Ibeo.

Abb. 5.3 zeigt wiederum einen Ist-Soll-Vergleich einer 3D-Erfassung mit einer $1/4^\circ$ Winkelauflösung. Als Messumgebung für diesen Vergleich diente wieder der Flur der Firma Sick-Ibeo. In der Abb. 5.3 ist zu sehen, wie einige Gegenstände der Messumgebung erkannt werden. Die Darstellung zeigt auch einige fehlerhafte 3D-Messpunkte. Ursache dieser Störungen können spiegelnde Oberflächen sein, z.B. die Flurbilder oder die Flurlampen. Andere Störungen entstanden durch Mitarbeiter, die während der Erfassung den Flur betreten haben.

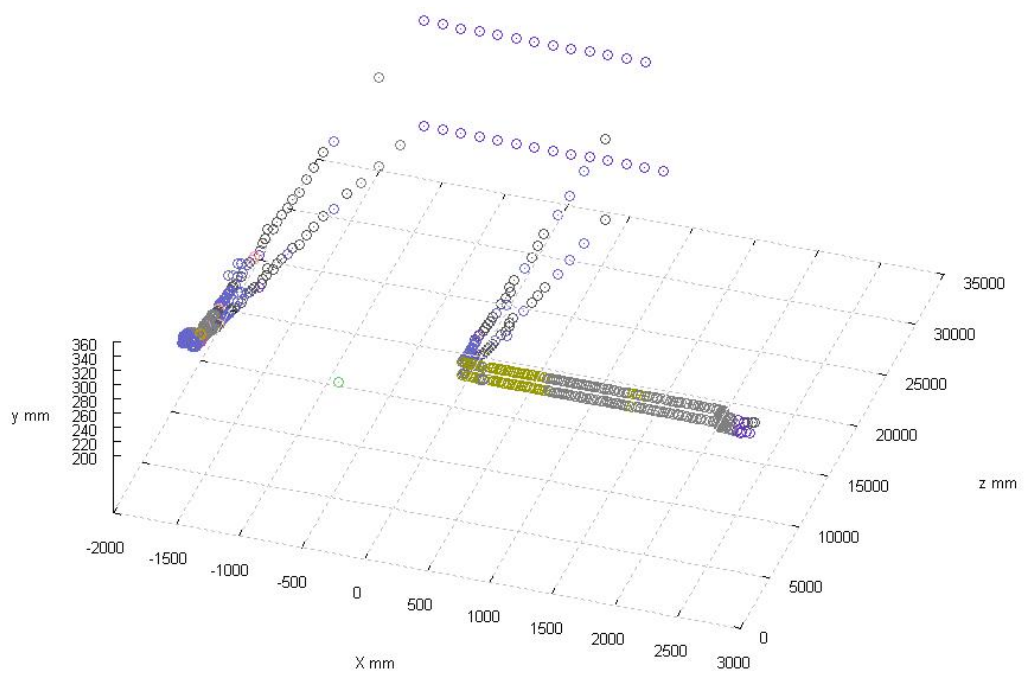


Abbildung 5.2.: Visualisierung zweier 3D-Datenprofile, die im Step-Modus nacheinander ermittelt wurden.

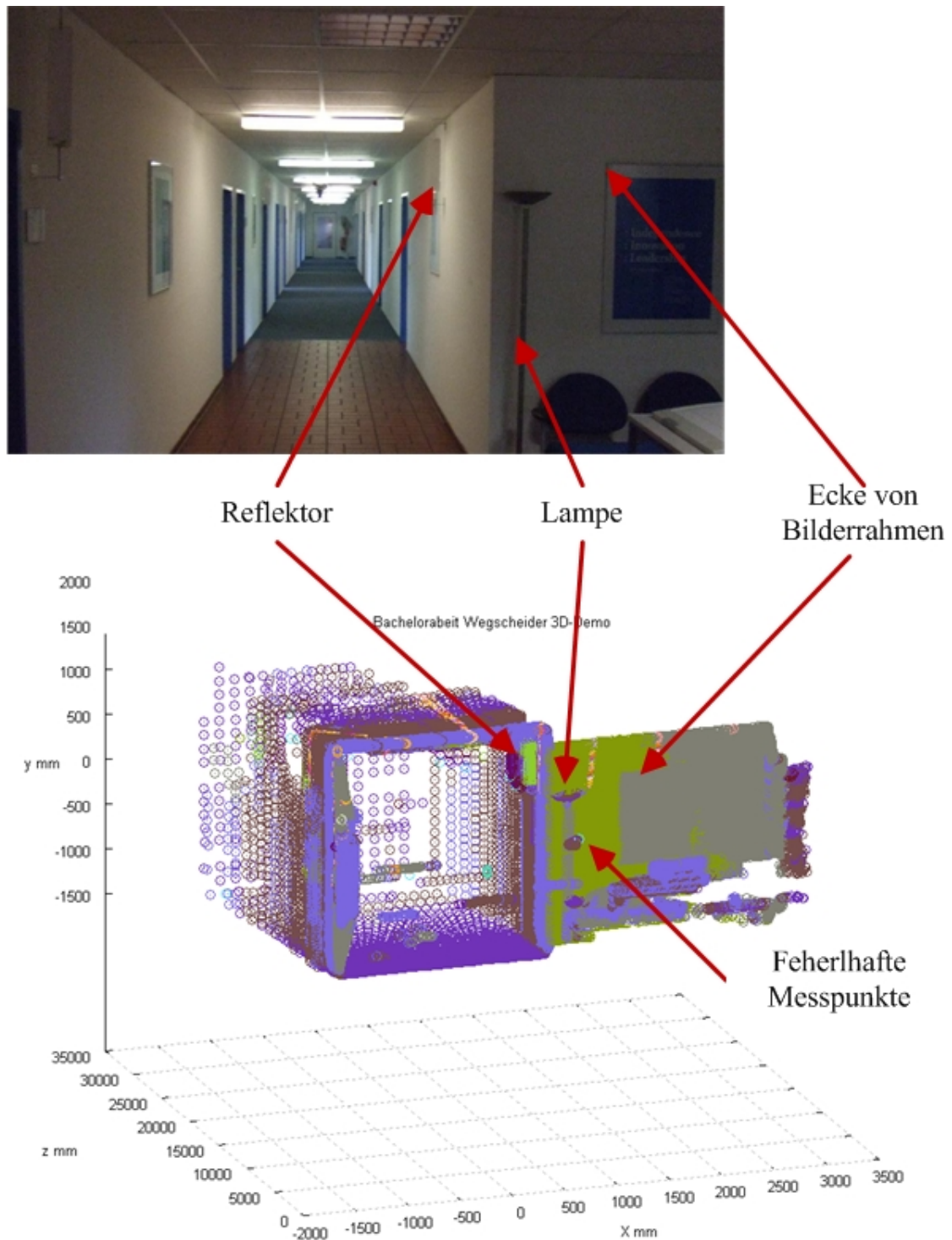


Abbildung 5.3.: Ist-Soll-Vergleich einer 3D-Erfassung mit einer Winkelauflösung von $1/4^\circ$.

Zwei aufgetretene Fehler, die nicht auf Anrieb behoben werden konnten, schränken jedoch die 3D-Erfassung ein. Beide Fehler sind auf Timing-Probleme in der Kommunikation zwischen Messrechner und der Erfassungssteuerung zurückzuführen und betreffen nur indirekt den Erfassungsalgorithmus.

Der eine Fehler entsteht, wenn ein 3D-Datenprofil aufgrund der Größe in zwei oder mehr Pakete aufgeteilt werden muss. Die Aufteilung in der Erfassungssteuerung funktioniert, jedoch können die aufgeteilten Datenpakete nicht direkt hintereinander versendet werden. Das erste Paket kam nur in Stücken oder gar nicht in der Testanwendung an. Vermutet wird, dass das zweite gesendete Paket den Sendebuffer des Messrechners überschreibt, bevor dieser das erste Paket komplett versendet hat. Wird das zweite Paket verzögert zum Anwender gesendet, simuliert durch einen Breakpoint, werden beide Pakete vollständig zur Testanwendung gesendet und dort erfolgreich verarbeitet. Das bedeutet, in der momentanen Version können nur 10240 Bytes Messdaten erfasst werden.

Der zweite Fehler tritt bei Erfassungen mit einer Winkelauflösung von $1/8^\circ$ auf, die in der Regel große 3D-Profile in kurzer Zeit ermitteln. Dauertests ergaben, dass die Erfassung mal nach kürzerer oder längerer Zeit abgebrochen wird. Die Erfassungssteuerung meldete jeweils ein Timeout in der Spi-Can-Bus-Kommunikation mit der Powercubeeinheit. Vermutet wird, dass der Messrechner die Erfassungssteuerung bei einer Kommunikation nicht rechtzeitig bedienen kann, da er noch damit beschäftigt ist, Daten zum Anwender zu senden.

Beide Fehler hätten durch eine Änderung der Firmware des Laserscanners behoben werden können, was jedoch aus Termin- und Prioritätsgründen nicht rechtzeitig möglich war.

Um eine zeitliche Betrachtung der Algorithmen des Step-Modus unter hoher Datenlast herzustellen, wurde eine 3D-Erfassung mit einer Winkelauflösung von $1/4^\circ$ und einen Scanbereich von 180° für den Laserscanner durchgeführt. Diese Konfiguration bedeuten, dass 720 3D-Messpunkte in einem 3D-Single-Scan verarbeitet (einschließlich der Transformation in das lokale Koordinatensystem der 3D-Einheit) und anschließend zum Anwender versendet werden. Mehr Datenvolumen konnte aufgrund der beiden zuvor beschriebenen Fehler nicht erzeugt werden.

Folgende Ergebnisse wurden, wie in Abb. 5.4 belegt wird, ermittelt:

- Eine 2D-Datenprofil-Erfassung, Errechnung einer neuen Position und die Erstellung und die Versendung eines 3D-Datenprofils (Schritte 2 bis 5 des Messvorgangs im Step-Modus vgl. Punkt 4.5.2) dauert ca. 170 ms.
- Die Datenverarbeitung ohne einen 2D-Scan(Schritte 3 bis 5 des Messvorgangs) dauert ca. 56 ms.

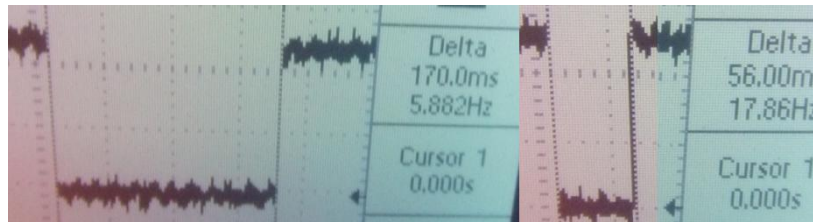


Abbildung 5.4.: Links: Dauer der Schritte 2 bis 5 des Messvorgangs im Step-Modus.
Rechts: Dauer der Schritte 3 bis 5 des Messvorgangs im Step-Modus.

Diese Messungen bedeuteten, dass die Datenverarbeitung im Step-Modus deutlich schneller durchgeführt wird, als eine Schrittbewegung, die wie in Kap. 4.5.2 bereits beschrieben, 300 ms dauert. Aus diesen Ergebnissen resultiert, dass eine Erfassung eines 3D-Single-Scans in der Erfassungssteuerung zeitlich betrachtet aus der Dauer eines 2D-Laserscans und einer Schrittbewegung zusammengesetzt ist. Das heißt, die Erfassungssteuerung kann ein 3D-Datenprofil in ca. 420 ms (300 ms Schrittbewegung + 120 ms 2D-Laserscan) erfassen.

Wie unter Punkt 4.3.3 erläutert wurde, wurde aufgrund eines Kommunikationsfehlers eine Ausführung eines 3D-Single-Scans an das Kommando „Receive-3D-Scan“ gebunden. Könnte dieser Kommunikationsfehler behoben werden, könnte ein 3D-Profil alle 420 ms + einem Offset für die TCP/IP-Übertragung zum Anwender übertragen werden.

5.4. Zusammenfassung

In diesem Kapitel wurde die Qualitätssicherung der Entwicklung der 3D-Erfassung dieser Arbeit besprochen. Es wurde dargestellt, welche Testsoftware erstellt wurde und wie der Test der 3D-Einheit durchgeführt wurde. Am Schluss erfolgte eine Zusammenfassung des Status der 3D-Einheit, nachdem die Qualitätssicherung abgeschlossen wurde.

6. Zusammenfassung und Ausblick

In der vorliegenden Arbeit konnte erfolgreich ein Prototyp realisiert werden, der eine 3D-Erfassung einer Messumgebung ermöglicht. Zwei Einheiten, ein 2D-Laserscanner und eine Motorservoschwenkeinheit (kurz Powercubeeinheit), werden zu einer logischen 3D-Einheit bzw. einer logischen 3D-Einheit verknüpft. Um eine 3D-Erfassung zu ermöglichen und einen Prototypen zu erstellen, wurden mehrere Module charakterisiert, entworfen und umgesetzt.

6.1. Bewertung der realisierten 3D-Erfassung

6.1.1. Erreichte Ziele

Es wurde eine Anwenderschnittstelle entworfen und umgesetzt, die alle Anforderungen, die unter Punkt 3.5.2 beschrieben wurden, erfüllt. Der Benutzer der 3D-Einheit kann über eine leicht bedienbare und verständliche Anwenderschnittstelle einen 3D-Laserscan konfigurieren, ausführen und die 3D-Daten entgegennehmen, ohne dass er Wissen über den internen Aufbau der 3D-Einheit benötigt. Die Anwenderschnittstelle wurde wiederum an eine State-Machine angebunden, die für eine zustandsabhängige Kommandoverarbeitung und Fehlerüberwachung sorgt. So wird ein geordneter Ablauf der 3D-Erfassung gesteuert bzw. Fehler, die durch falsche Eingaben des Benutzers oder im internen Ablauf ausgelöst werden, werden erkannt und ausgewertet. Beide Module, die eng miteinander arbeiten, können leicht erweitert und angepasst werden und könnten als Grundlage für andere 3D-Entwicklungen dienen.

Mit dem Modul Step-Modus konnte eine Variante der Synchronisation der Bewegungen der Powercubeeinheit und des 2D-Laserscanners entworfen und umgesetzt werden. Die Erfassungssteuerung kann zuverlässig in kurzer Zeit 3D-Datenprofile mit Winkelauflösungen von 1° , $1/2^\circ$ und $1/4^\circ$ erfassen.

Im Modul 3D-Transformation werden die synchronisierten Daten der beiden Einheiten verarbeitet, indem jeweils eine 2D-Koordinate des Laserscanners und der

Positionswert der Powercubedrehachse zu einer 3D-Koordinate transformiert wird. Die Transformation erfolgt in das lokale Koordinatensystem der Einheit, welches in dieser Arbeit festgelegt wurde. Weiter wurden mit dem Modul Grundlagen für weitere 3D-Projekte geschaffen. Um die Transformation durchführen zu können, wurden mehrere 3D-Basistransformationen umgesetzt, die für weitere 3D-Projekte wieder verwendet werden können.

6.1.2. Offene Punkte und Verbesserungen

Eine Verbesserung der 3D-Erfassung kann mit der Realisierung des Interpolations-Modus erfolgen. Wird der Interpolations-Modus erstellt, wie er geplant ist, dann wird im Gegensatz zum Step-Modus die Parallelität des Messrechners, der Powercubeeinheit und der Erfassungssteuerung noch effizienter ausgenutzt. Durch die komplette parallele Verteilung der Datenermittlung auf der Powercubeeinheit und dem Messrechner auf der einen Seite und der Datenverarbeitung in der Erfassungssteuerung auf der anderen Seite, könnte die Dauer einer Erfassung eines 3D-Datenprofils weiter verkürzt werden und ein 3D-Laserscan könnte schneller eine Messumgebung abtasten.

Um den Entwurf des Interpolations-Modus umsetzen zu können, müssen die unter Kap. 4.6.3 beschriebenen offenen Punkte zuerst gelöst werden.

Eine weitere Verbesserung der Erfassung im Step-Modus bzw. eine Notwendigkeit für die Realisierung des Interpolations-Modus ist die Verbesserung der Firmware des Messrechners. Könnte das unter Kap. 4.3.3 beschriebene Problem bei der gleichzeitigen Kommunikation zwischen der Erfassungssteuerung mit dem Anwender und der Powercubeeinheit behoben werden, dann wäre die Ausführung eines 3D-Single-Scans nicht mehr an das Kommando „Receive-3D-Scan“ gebunden. In diesem Fall wäre eine Ausführung eines 3D-Single-Scans nicht vom Anwender abhängig und die 3D-Datenprofile könnten selbständig erfasst und zum Anwender versendet werden. Die Dauer für eine Erfassung würde sich zeitlich an die Ausführung des Step-Modus-Vorgangs orientieren, der unter Kap. 5.3 zeitlich betrachtet wurde.

Werden weiter die in Kap. 5.3 beschriebenen Fehler im Timing der Kommunikation zwischen Messrechner und Erfassungssteuerung behoben, wird erwartet, dass ein 3D-Laserscan mit einer Winkelauflösung von $1/8^\circ$ problemlos ausgeführt werden kann.

6.2. Erweiterungsmöglichkeiten

Speichern von Daten in den Flash des Laserscanners:

Die 3D-Datenerfassung könnte um eine Flashansteuerung erweitert werden. Ein generelles Speichern von Daten in den Flash des Laserscanners ist bereits über den Messrechner möglich. Die Basefunction stellen dazu bereits Funktionen zur Verfügung [vgl. Kap.7.5 Sick Ibeo GmbH, 2005]. Entworfen werden müsste noch eine Struktur, wie welche Daten im Flash gespeichert werden könnten. Die Konfigurationsdaten eines 3D-Laserscans ließen sich z.B. dadurch permanent speichern und könnten nach Wiedereinschalten der 3D-Einheit in der 3D-Erfassungssteuerung geladen werden.

Einbau von Filtern in die 3D-Datenerfassung:

Eine sinnvolle Erweiterung für die 3D-Erfassung wäre der Einbau von Filtern, die vorwiegend aus der Bildverarbeitung übernommen werden könnten. Die Filter könnten gezielt Störungen, die durch spiegelnde Oberflächen oder Regen verursacht werden, identifizieren und ausblenden. Ein Medienfilter [vgl. Wikipedia.org, 2008a] könnte z.B. eingesetzt werden, um einen in der Distanz fehlerhaften gemessenen 2D-Wert zu „glätten“. Weiterhin könnten Filter genutzt werden, um eine Mustererkennung in den 3D-Daten zu erleichtern.

Einige Filter wurden bereits für andere Projekte in der Firma entwickelt. Zu prüfen ist, wie weit diese Filter für die 3D-Datenerfassung dieser Arbeit übernommen werden können.

Literaturverzeichnis

- [Bertagnolli 2004] BERTAGNOLLI, Frank: Vollautomatische robotergestützte 3D-Oberflächenvermessung zur Qualitätssicherung von Werkstücken. In: VDI/VDE-GESELLSCHAFT MESS- UND AUTOMATISIERUNGSTECHNIK (Hrsg.): *Sensoren und Messsysteme 2004 : Tagung Ludwigsburg, 15. und 16. März 2004*, VDI-Verlag Düsseldorf, März 2004, S. 801–804. – ISBN 3-18-091829-2
- [Borowiack 2006] BOROWIACK, Malte: Neues Ultraschallgerät liefert schärfere Bilder von Kinderherzen; MEDIZIN: Uni-Kinderklinik zeigt Untersuchung. In: *General-Anzeiger Bonn* (2006). – Rubrik Medizin S.4
- [CAN in Automation(CIA) 2008] CAN IN AUTOMATION(CIA): *Controller Area Network (CAN)*. 2008. – URL <http://www.can-wiki.info/%20und%20Downloads%20CAN%20Specification%202.0>. – Zugriffsdatum: 15.03.2008
- [Directindustry 2007] DIRECTINDUSTRY ; DIRECTINDUSTRY (Hrsg.): *3D laserscanner*. 2007. – URL <http://www.directindustry.de/tab/3D-laserscanner.html>. – Zugriffsdatum: 19.12.2007. – Übersicht über Angebote für 3D-Scanner.
- [Eichler 2007] EICHLER, Jürgen: *Grundlagen für das Ingenieurstudium - kurz und prägnant*. 3. Friedrich Vieweg und Sohn | GWV Fachverlage GmbH, 2007. – ISBN 978-3-8348-0223-1
- [Elkhalili 2005] ELKHALILI, Omar: *Entwicklung von optischen 3D CMOS-Bildsensoren auf Basis der Pulslaufzeitmessung*, Universität Duisburg-Essen, Dissertation, 2005. – URL http://www.ub.uni-duisburg.de/ETD-db/theses/available/duett-05072005-130542/unrestricted/Elkhalili_Diss.pdf. – Zugriffsdatum: 19.12.2007. – Elektronische Quelle. Hinterlegt auf der der beigelegten CD.
- [Faber 2007] FABER, Jürgen ; SICK AG (Hrsg.): *Praxissemesterarbeit 3D-Scan im Sick-Standort Reute*. 2007. – Die Praxissemesterarbeit wurde in der Entwicklungsabteilung bei Sick im Standort Reute angefertigt. Sie besitzt einen Sperrvermerk für eine außerbetriebliche Weitergabe und ist daher nicht auf der hinterlegt

- [Faro Europe 2007] FARO EUROPE ; FARO EUROPE (Hrsg.): *Homepage Firma FARO EUROPE GmbH und Co. KG*. 2007. – URL <http://www.faro.com/content.aspx?ct=ge&content=pro&item=5>. – Zugriffsdatum: 22.05.2008
- [Fohl 2007] FOHL, Wolfgang: *Verhaltensbeschreibung reaktiver Systeme*. 2007. – URL <http://users.etech.haw-hamburg.de/users/fohl/pl/wse6harel.pdf>. – Zugriffsdatum: 13.05.2008
- [Gühring 2002] GÜHRING, Jens: *3D-Erfassung und Objektkonstruktion mittels Streifenprojektion*, Universität Stuttgart, Dissertation, 2002. – URL http://deposit.ddb.de/cgi-bin/dokserv?idn=980546680&dok_var=d1&dok_ext=pdf&filename=980546680.pdf. – Zugriffsdatum: 19.12.2007. – Elektronische Quelle. Hinterlegt auf der der beigelegten CD.
- [Gnuplot 2008] GNUPLOT ; GNUPLOT (Hrsg.): *Homepage des Gnuplot-Plotters*. 2008. – URL <http://www.gnuplot.info/>. – Zugriffsdatum: 22.05.2008. – Version 4.2.3 wurde eingesetzt.
- [Greuer 2007] GREUER, Andreas: *CAIC 3D PRO*. 2007. – URL <http://www.calc3d.com/gindex.html>. – Zugriffsdatum: 15.12.2007
- [Grieger 1992] GRIEGER, Ingolf: *Graphische Datenverarbeitung. Mit einer Einführung in Phigs und Phigs-Plus*. 2. Springer-Berlin, 1992. – ISBN 3-540-53747-3
- [Harel 1998] HAREL, David: *Modeling reactive systems with statecharts : the state-mate approach*. McGraw-Hill New-York[u.a.], 1998. – ISBN 0-07-026205-5
- [Hugenschmidt 2007] HUGENSCHMIDT, Manfred: *Lasermesstechnik, Diagnostik der Kurzzeitphysik*. Springer-Berlin, 2007. – ISBN 978-3-540-29920-2
- [J.Pont 2001] J.PONT, Michael: *Patterns for Time-Triggered Embedded Systems*. Addison Wesley New-York[u.a.], 2001. – ISBN 0-201-33138-1
- [Kern 2008] KERN, Thomas ; 3DSOURCE.DE (Hrsg.): *Infos rund um die 3D-Computergrafik, OpenGL und VRML*. 2008. – URL <http://www.3dsource.de/deutsch/3Dmathe.htm>. – Zugriffsdatum: 15.01.2008
- [Lase-GmbH 2007] LASE-GMBH ; LASE-GMBH (Hrsg.): *Homepage Firma Lase GmbH*. 2007. – URL <http://www.lase.de/de.html>. – Zugriffsdatum: 19.12.2007
- [L.Paul 2008] L.PAUL: *Vorlesung von Univ.-Prof. Dr.rer.nat. L. Paul zu 3D-Verarbeitung*. 2008. – URL <http://www2.uni-wuppertal.de/fb8/techphy/Paul/Lehre/Chapter3.ppt>. – Zugriffsdatum: 15.03.2008. – Elektronische Quelle: Leider lässt sich nicht mehr zuordnen zu welcher Vorlesung diese Präsentation erstellt wurde, weil teilweise keine Links mehr vorhanden waren.

- [Riegl 2007] RIEGL ; RIEGL (Hrsg.): *Homepage Firma Riegl*. 2007. – URL <http://www.riegl.co.at/>. – Zugriffsdatum: 22.05.2008
- [Schunk GmbH 2007a] SCHUNK GMBH: *Datenaustausch mit PowerCube*, 2007. – Elektronische Quelle. Beschreibung der PowerCube-Kommunikationsschnittstellen. Neuste Version nur auf Anfrage erhältlich. Hinterlegt auf der der beigelegten CD.
- [Schunk GmbH 2007b] SCHUNK GMBH ; SCHUNK GMBH (Hrsg.): *Programmieranleitung PowerCube*. 2007. – Elektronische Quelle. Neuste Version nur auf Anfrage erhältlich. Hinterlegt auf der der beigelegten CD.
- [Schunk GmbH 2008] SCHUNK GMBH: *Drehmodule elektrische Antriebe Part3*, 2008. – URL http://www.de.schunk.com/schunk_files/attachments/Drehmodule_Part3_1107__280_464__DE.pdf. – Zugriffsdatum: 29.02.2008. – Elektronische Quelle. Neuste Version nur auf Anfrage erhältlich. Hinterlegt auf der der beigelegten CD.
- [Sick AG 2007] SICK AG ; SICK AG (Hrsg.): *Homepage Firma Sick AG*. 2007. – URL <http://www.sick.de>. – Zugriffsdatum: 19.12.2007
- [Sick Ibeo GmbH 2004a] SICK IBEO GMBH: *Lasermesssystem LD-OEM Betriebsanleitung*, 2004. – URL <http://www.mysick.com/saqqara/pdf.aspx?id=im0011750>. – Zugriffsdatum: 02.02.2008. – Elektronische Quelle. Hinterlegt auf der der beigelegten CD.
- [Sick Ibeo GmbH 2004b] SICK IBEO GMBH: *Lasermesssystem LD-OEM:Ergänzung zur Betriebsanleitung:Beschreibung und Technische Daten des LD-LRS1000/2100/3100*, 2004. – elektronische Quelle. Nur Auf Anfrage erhältlich. Hinterlegt auf der der beigelegten CD.
- [Sick Ibeo GmbH 2005] SICK IBEO GMBH: *Programming Guide C32 Application Development*, 2005. – elektronische Quelle. Dokument per Anfrage erhältlich. Hinterlegt auf der der beigelegten CD.
- [Sick Ibeo GmbH 2006] SICK IBEO GMBH: *User Protocol Services for Operating/Configuring the LD-OEM/LD-LRS Laser Measurement System*, 2006. – URL <http://www.mysick.com/saqqara/pdf.aspx?id=im0017890>. – Zugriffsdatum: 02.02.2008. – Elektronische Quelle. Hinterlegt auf der der beigelegten CD.
- [Texas Instrument 2004] TEXAS INSTRUMENT: *TMS320C3x User's Guide*, 2004. – URL <http://focus.ti.com/lit/ug/spru031f/spru031f.pdf>. – Zugriffsdatum: 28.02.2008. – elektronische Quelle. Hinterlegt auf der der beigelegten CD.

- [The IEEE, Inc 1985] THE IEEE, INC: *IEEE Standard for Binary Floating-Point Arithmetic*. 1985. – URL <http://754r.ucbtest.org/standards/754.pdf>. – Zugriffsdatum: 15.03.2008
- [Wikipedia.org 2008a] WIKIPEDIA.ORG: *Bildverarbeitung*. 2008. – URL http://de.wikipedia.org/wiki/Bildverarbeitung#Nichtlineare_Filter. – Zugriffsdatum: 30.05.2008. – Filter für die 3D-Datenverarbeitung.
- [Wikipedia.org 2008b] WIKIPEDIA.ORG: *Interlace*. 2008. – URL <http://en.wikipedia.org/wiki/Interlace>. – Zugriffsdatum: 15.03.2008
- [Wikipedia.org 2008c] WIKIPEDIA.ORG: *Serial Peripheral Interface*. 2008. – URL http://de.wikipedia.org/wiki/Serial_Peripheral_Interface. – Zugriffsdatum: 15.03.2008
- [Wikipedia.org 2008d] WIKIPEDIA.ORG: *TCP.2FIP-Referenzmodell*. 2008. – URL <http://de.wikipedia.org/wiki/TCP/IP-Referenzmodell#TCP.2FIP-Referenzmodell>. – Zugriffsdatum: 15.03.2008

A. Memory Mapping einer C32-Anwendung im Hello-World-Zustand

name	origin	length	used	attributes
SRAM_INT	00001000	000000040	00000000	RWIX
STARTRAM	00001040	000000001	00000001	RWIX
LINKRAM	00001041	000000100	0000009e	RWIX
SRAM0	00001141	00001deff	00001c75	RWIX
BSSRAM	00020000	000020000	0000d434	RWIX
RAM0	00809800	000000400	00000000	RWIX
RAM1	00809c00	000000400	00000000	RWIX

Tabelle A.1.: Memory-Mapping einer Hello-World-C32-Anwendung mit Spi-Treiber und Interface-Funktionen.

B. Errorcode-Tabelle der 3D-Erfassungssteuerung

#define ERR-BASE	0xC000
#define ERR-3D-SUCCESS	0x0
#define ERR-3D-NO-CMD	(ERR-BASE + 0x01)
#define ERR-3D-CAN-NOT-READ-CMD	(ERR-BASE + 0x02)
#define ERR-3D-UNKNOWN-SCAN-MODE	(ERR-BASE + 0x03)
#define ERR-3D-NOT-VALID-SCAN-AREA	(ERR-BASE + 0x04)
#define ERR-3D-NO-CONFIG-EXIST	(ERR-BASE + 0x05)
#define ERR-3D-POWER-CUBE-ERROR	(ERR-BASE + 0x06)
#define ERR-3D-LD-ERROR	(ERR-BASE + 0x07)
#define ERR-3D-REQUEST-SCAN-FAILED	(ERR-BASE + 0x08)
#define ERR-3D-RECEIVE-SCAN-FAILED	(ERR-BASE + 0x09)
#define ERR-3D-INVALID-PARAM	(ERR-BASE + 0x0A)
#define ERR-3D-INVALID-PROFIL	(ERR-BASE + 0x0B)
#define ERR-3D-CONFIG-FAILED	(ERR-BASE + 0x0C)
#define ERR-3D-INIT-POWERCUBE-FAILED	(ERR-BASE + 0x0D)
#define ERR-3D-INIT-LD-SCANNER-FAILED	(ERR-BASE + 0x0E)
#define ERR-3D-MEASURE-ERROR	(ERR-BASE + 0x0F)
#define ERR-3D-CANCEL-SCAN-FAILED	(ERR-BASE + 0x10)
#define ERR-3D-SEND-DATA-FAILED	(ERR-BASE + 0x11)
#define ERR-3D-SET-LMS-MODE-FAILED	(ERR-BASE + 0x12)
#define ERR-3D-CREATE-ACK-FAILED	(ERR-BASE + 0x13)
#define ERR-3D-INVALID-COMMAND	(ERR-BASE + 0x14)

Tabelle B.1.: Errorcodetabelle der 3D-Einheit

C. Berechnung der Positionsabweichung der Powercubdrehachse im Step-Modus

Abkürzung	Bedeutung
AS_INC_F	Winkelschrittweite in Inkrementen (Float-Format)
AS_INC_I	Winkelschrittweite in Inkrementen (Integer-Format)
SP_INC_F	PCube-Start-Pos. im Step-Modus in Inkrementen (Float-Format)
SP_INC_I	PCube-Start-Pos. im Step-Modus in Inkrementen (Integer-Format)
Delta_AS_INC	Abweichung zwischen AS_INC und AS_INC_INT_CAST
Delta_SP_INC	Abweichung zwischen SP_INC und SP_INC_INT_CAST
Num_3D_SS	Anzahl an 3D-Single-Scans für einen 3D-Scan
Delta_Max	Maximale Abweichung durch die Umrechnung in Inkrementen

Formeln zur Berechnung der Positonsabweichung:

$$\Delta_{Max} = \Delta_{ANGLE_STEP_INC} \cdot Num_3D_SS + \Delta_{SP_INC}$$

Beispiel: Scanbereich Powercube (Maximaler Scanbereich 90° und Winkelauflösung 1°)

$$PCube_Start_Pos = -45^\circ // -201250 \text{ Inkrements}$$

$$PCube_End_Pos = 45^\circ // 201250 \text{ Inkrements}$$

$$Num_3D_SS = 90$$

$$\Delta_{AS_INC} = AS_INC_F - AS_INC_I = 4722,22 - 4722 = 0,22$$

$$\Delta_{SP_INC} = SP_INC_F - SP_INC_I = (-201250) - (-201250) = 0$$

$$Angle_Step = 1^\circ // 4722,22 \text{ Inkrements}$$

$$\Delta_{Max} = 0,22 \cdot 90 + 0 = 20 \text{ Inkrements} \Rightarrow 0,0045^\circ$$

D. 3D-Kommandos der Anwenderschnittstelle

Kommando	Get-State	
Beschreibung	Statusabfrage der 3D-Einheit	
Service-ID	0x1401	
Übergabe:		
Parametername	Größe/Datentyp	Parameterbeschreibung
-	-	-
Antwort (Acknowledge):		
Parametername	Größe/Datentyp	Parameterbeschreibung
WORKING-STATE	4 Bytes (char)	0x00 = START 0x01 = SYSTEM_ERROR 0x02 = IDLE 0x03 = 3D_MEASURE Statusbeschreibung s. 4.3.3
GLOBAL-ERROR-STATE	4 Bytes (short)	Globaler Error-Status (s. Anhang B)
LOKAL-ERROR-STATE	4 Bytes (short)	Lokaler Error-Status (s. Dokumentation der Einheiten unter Tab. 3.2)

Kommando	Set-3D-Config	
Beschreibung	Konfiguration eines 3D-Laserscans	
Service-ID	0x1402	
Übergabe:		
Parametername	Größe/Datentyp	Parameterbeschreibung
SCAN-MODE	4 Bytes (char)	1: Step-Mode 2: Interpolation
ANGLE-STEP	4 Bytes (char)	0x00 = 1° 0x01 = 1/2° 0x02 = 1/4° 0x04 = 1/8°
START-ANGLE-HORIZONTAL	4 Bytes (short)	START-ANGLE Laserscanner: Bereich: 0° - 180° Angabe in 1/16° Format (Winkelformat des Laserscanners [s. Kap.5.3 Sick Ibeo GmbH, 2006])
END-ANGLE-HORIZONTAL	4 Bytes (short)	END-ANGLE Laserscanner: Wertebereich: 0° - 180° Angabe in 1/16° Format Bed. End- >Startwinkel
START-ANGLE-VERTIKAL	4 Bytes (short)	START-ANGLE Powercubeinheit: Bereich: 0 - 90° Angabe in 1/16° Format
END-ANGLE-VERTIKAL	4 Bytes (short)	END-ANGLE Powercubeinheit: Angabe wie START-ANGLE-VERTIKAL Bed.End- ≥ Startwinkel
3D-PROFIL	4 Bytes (char)	0x00 = XYZ-Punkte 0x01 = Rohdaten in Polarformat (s. Kap. 4.2.2)
Antwort (Acknowledge):		
Parametername	Größe/Datentyp	Parameterbeschreibung
-	-	-

Kommando	Get-3D-Config	
Beschreibung	Lesen der 3D-Konfigurationsdaten.	
Service-ID	0x1403	
Übergabe:		
Parametername	Größe/Datentyp	Parameterbeschreibung
-	-	-
Antwort (Acknowledge):		
Parametername	Größe/Datentyp	Parameterbeschreibung
Parameter sind identisch dem Übergabeparameter des Kommandos Set 3D-Config		

Kommando	Receive-3D-Scan	
Beschreibung	Holt ein 3D-Datenprofil eines 3D-Single-Scans ab.	
Service-ID	0x1408	
Übergabe:		
Parametername	Größe/Datentyp	Parameterbeschreibung
-	-	-
Antwort (Acknowledge):		
Parametername	Größe/Datentyp	Parameterbeschreibung
Die Rückgabe ist ein Datenprofil. Für den Aufbau eines Datenprofils s. Kap. 4.2.2		

Kommando	Set-State-xxx	
Beschreibung	Setzt einen Zustand der 3D-Erfassungssteuerung. Hinweise: In Zustand 3D_MEASURE kann nur gewechselt werden, wenn vorher die 3D-Erfassungssteuerung erfolgreich konfiguriert wurde. (vgl. 4.3.3)	
Service-ID	SET IDLE	0x1404
Service-ID	SET 3D_MEASURE	0x1405
Übergabe:		
Parametername	Größe/Datentyp	Parameterbeschreibung
-	-	-
Antwort (Acknowledge):		
Parametername	Größe/Datentyp	Parameterbeschreibung
-	-	-

E. Inhalt der CD

Dieser Arbeit liegt eine CD-ROM mit folgendem Inhalt bei:

Bachelorarbeit Wegscheider.pdf Diese Arbeit im PDF-Format.

Dokumentationen zur Powercubeinheit:

- Kommunikation: Datenaustausch mit PowerCube.pdf [Schunk GmbH, 2007a].
- Funktionen: Programmieranleitung PowerCube.pdf [Schunk GmbH, 2007b].
- Technische Details: Drehmodule_Part3_DE.pdf [Schunk GmbH, 2008].

Dokumentationen zum 2D-Laserscanner:

- LD_OEM_Betriebsanleitung.pdf [Sick Ibeo GmbH, 2004a].
- Ergänzung_zur_Betriebsanleitung.pdf [Sick Ibeo GmbH, 2004b].
- TMS320C3x_User's_Guide [Texas Instrument, 2004].
- C32-APPLICATION_DEVELOPMENT_13_E.pdf [Sick Ibeo GmbH, 2005].
- User Protocol Services for Operating_LD-OEM1000_LRS.pdf [Sick Ibeo GmbH, 2006]

Harelautomat: wse6harel.pdf [Fohl, 2007].

Sonstige elektronische Quellen:

- 3D-Erfassung_und_Objekt konstruktion_mittels_Streifenprojektion.pdf [Gühring, 2002].
- Entwicklung vonoptischen 3D-CMOS Bildsensoren auf Basis der Pulslaufzeitmessung.pdf [Elkhalili, 2005].

Versicherung über Selbstständigkeit

Hiermit versichere ich, dass ich die vorliegende Arbeit im Sinne der Prüfungsordnung nach §24(5) ohne fremde Hilfe selbstständig verfasst und nur die angegebenen Hilfsmittel benutzt habe.

Hamburg, 25. Juni 2008

Ort, Datum

Unterschrift