

Navigation eines autonomen Roboters mittels Lasertriangulation

Stephan Edel

edel@informatik.hu-berlin.de

Humboldt-Universität zu Berlin

Mathematisch-Naturwissenschaftliche Fakultät II

Institut für Informatik

3. Juni 2010

Nachbearbeitete Fassung der am 30. April 2010 eingereichten Diplomarbeit.

Gutachter:

Prof. Dr. rer. nat. Ralf Reulke

Prof. Dr. Hans-Dieter Burkhard.

Inhaltsverzeichnis

Abstract	V
Abkürzungsverzeichnis	VI
Symbolverzeichnis	VIII
1 Einleitung	1
1.1 Motivation	1
1.2 Forschungsstand	2
1.2.1 Autonome Fahrzeuge	2
1.2.2 Entfernungssensoren	3
1.3 Ziel der Arbeit	5
2 Grundlagen	6
2.1 Kamerakalibrierung nach Zhang	8
3 Umfelderkennung	10
3.1 Funktionsprinzip	11
3.2 Kalibrierungsverfahren	12
3.2.1 Innere Orientierung der Kamera	13
3.2.2 Relative Orientierung von Kamera und Laser	14
3.2.3 Äußere Orientierung des Sensors	19
3.3 Implementierung	22
3.3.1 Mechanische Konfiguration	22

3.3.2	Automatisierte Kalibrierung	23
3.3.3	Bilderfassung	25
3.3.4	Bildverbesserung	27
3.3.5	Peakdetektion	27
	Subpixel-Genauigkeit	30
3.3.6	Hinzufügen von Zufallswerten	32
3.4	Ergebnisse	32
3.4.1	Nahbereich	32
3.4.2	Weitbereich	36
3.4.3	Bewertung	36
4	Autonomes System	39
4.1	Systemarchitektur	39
4.1.1	Hardware	39
4.1.2	Software	42
4.1.3	Odometrie	43
4.2	Weltmodell	44
4.2.1	Bereiche von Interesse	45
4.2.2	Indikatoren	47
4.2.3	Navigationskarten	49
4.2.4	Wegsuche	50
4.2.5	Umgang mit Odometriefehlern	51
4.2.6	Lokalisierung	51
	Kartenupdate	51
	Drehkorrektur	52
4.3	Ergebnisse	55
4.3.1	Odometriebasierte Navigation	55
	Erste autonome Fahrt	55
	Zweite autonome Fahrt	55
4.3.2	Korrigierte Navigation	56

5	Fazit	57
6	Ausblick	58
	Anhang	60
A	Laserfokussierung	60
B	Verwendete Software	60
C	Hardwarekomponenten	61
D	Betriebsanleitung ARP-117	62
	Inhaltsverzeichnis des Datenträgers	68
	Danksagung	71

Abstract

For this thesis, an autonomous mobile robot was designed, built, and tested. As primary means of perception, a light-stripe 3D sensor was devised, and a method is proposed to calibrate this sensor. Furthermore, a method is proposed to identify unknown areas which have yet to be discovered. Using these assets, the robot is able to navigate unknown indoor environments and draw a map of obstacles and accessible areas. With minor exceptions, only free software and consumer-market hardware components were used in the device and the design process.

Abkürzungsverzeichnis

2D	zweidimensional
3D	dreidimensional
ARP-117	das in dieser Arbeit entwickelte autonome System
CCD	<i>Charge-Coupled Device</i> , hier:ameratechnologie
CMOS	komplementärer Metall-Oxid-Halbleiter; hier:ameratechnologie
CPU	Hauptprozessor
EM	elektromagnetisch
FOV	<i>Field of View</i> : Gesichtsfeld
GPS	<i>Global Positioning Network</i>
I/O	Eingabe / Ausgabe
JPEG	<i>Joint Photographic Experts Group</i> , hier: Bildformat
LIDAR	<i>Light Detection and Ranging</i>
LUT	<i>Lookup-Table</i>
MC	<i>Mikrocontroller</i>
MOSFET	Metall-Oxid-Halbleiter-Feldeffekttransistor
PC	<i>Personal Computer</i>
RAM	<i>Random Access Memory</i>
ROI	<i>Region of Interest</i>
SLAM	<i>Simultaneous Localization and Mapping</i>
S/N	Signal/Rausch-Verhältnis
S/W	Schwarz/Weiss
SSH	<i>Secure Shell</i> , ein Netzwerkprotokoll

SSID	<i>Service Set Identifier</i> , Kennung eines Drahtlosnetzwerks
STL	<i>Stereo Lithography</i> , steht hier für ein Dateiformat
TOF	<i>time-of-flight</i> , Laufzeit eines EM-Signals
UART	<i>Universal Asynchronous Receiver Transmitter</i>
USB	<i>Universal Serial Bus</i>
WLAN	<i>Wireless Local Area Network</i> : lokales, funkbasiertes Computernetzwerk

Symbolverzeichnis

Dies sind vor allem abschnittsübergreifend verwendete Symbole; lokal verwendete Symbole sind nicht vollständig aufgeführt.

B	Bildebene
c	Kamerahauptpunkt
C	Kameraprojektionszentrum
EH	Einheit des 3D-Kamerakoordinatensystems, implementiert als 3.50 cm
$K = (p_1, p_2, k_1, k_2)$	Parameter der Kameraverzerrung
L	Laserebene
L_i	Schnittgerade von L und S_i
$LUT : p \rightarrow P$	Lookup-Table
M	Fundamentalmatrix der Kamera
MA	Hochachse des Scanners in der Achse des Schrittmotors
M_f	2D-Karte des befahrbaren Bereichs
M_w	2D-Karte der Hindernisse
$p = (x, y)$	allg. Bildpunkt
$P = (X, Y, Z)$	allg. Punkt im Raum
S_i	Ebene des Kalibrierungsmusters i
Φ	Rotation des Scanners um MA

1 Einleitung

1.1 Motivation

Robotersysteme werden seit Jahrzehnten in vielen Bereichen eingesetzt. Dabei unterscheiden sich autonome mobile Roboter mehrfach von anderen Robotersystemen. Sie unterscheiden sich von bloßen Teleoperatoren; sie unterscheiden sich aber auch von Systemen mit festen oder verzweigten Sequenzen von Bewegungsabläufen wie dem klassischen, nicht mobilen industriellen Manipulator. Auch dieser verfügt meist über eine Sensorik, die in seinem Arbeitsbereich ungeplant auftretende Objekte feststellen kann; das Auftreten eines solchen Objekts wird dabei meist als Störung behandelt und führt zu einer Unterbrechung des Programmablaufs und der Aktivierung von Sicherheitsroutinen, die einen Schaden an Gerät oder Umgebung verhindern sollen. Der Aufwand einer Implementierung eines darüber hinausgehenden Umgangs mit unbekannten Faktoren ist in der Regel nicht verhältnismäßig. Auch die meisten heute eingesetzten mobilen Robotersysteme arbeiten nach diesem Prinzip, z.B. Systeme zur automatisierten Lagerhaltung. Diese werden mit einem vollständigen Modell ihres Einsatzbereichs versehen, oder es wird die Umgebung selbst für einen Einsatz dieser Systeme angepasst, z.B. durch optische Hilfsmarkierungen oder magnetische Leitschienen. Ein Beispiel für eine Anwendung in einer definierten Domäne ist der Roboterfußball, wo bei der Programmierung der vorwiegend optischen Sensoren mit einigen Annahmen über die Domäne gearbeitet werden kann, wie zum Beispiel mit der Farbe des Balls (Edel, 2009) oder der Anordnung der Feldlinien. Demgegenüber ist das Bereitstellen vollständiger Umgebungspläne oder gar eine vorbereitende Modifikation der Umgebung für viele Einsatzbereiche nicht praktikabel.

Dies gilt beispielsweise für autonome Systeme, die am allgemeinen Straßenverkehr teilnehmen sollen; Systeme, die selbstständig eine Erkundung zerstörter Gebäude nach Katastrophenfällen vornehmen (RoboCup Rescue); oder solche, die zur Beseitigung von Landminen verwendet werden. Aber auch im Consumer- und Haushaltsbereich ist es dem Endanwender kaum zuzumuten, eine präzise Kartierung seiner Wohnung oder seines Büros vorzunehmen.

1.2 Forschungsstand

1.2.1 Autonome Fahrzeuge

Im Haushaltsbereich verbreitet sind autonome Staubsauger wie z.B. der Roomba. Diese orientieren sich mit einfachen Infrarotsensoren, bei dem von Infrarotdioden emittiertes Licht von nahen Objekten reflektiert und von Photozellen registriert wird. Diese Systeme erstellen keine Karten ihrer Umgebung und navigieren vor allem reaktiv; zur Navigation mehrerer Räume werden beidseitig von Türdurchgängen Infrarot-Sender aufgestellt, an denen sich das Fahrzeug orientieren kann.

Im militärischen Bereich ist die autonome Navigation fliegender Systeme recht weit fortgeschritten; es wird dort mittlerweile eine solche Vielzahl von Geräten eingesetzt, dass Bedarf nach einer systematischen Klassifizierung ihrer Autonomie besteht (Alexander u. a., 2007); bei bodengebundenen Systemen mit militärischer Nutzung herrscht dagegen noch die Fernsteuerung vor (z.B. „Dragon Runner“, Schempf u. a. 2003).

Als Beispiel für eine zivilere Nutzung mobiler Roboter sei ein Forschungsprojekt von Thrun u. a. (2003) genannt. Diese entwickelten ein System, das mit Hilfe eines TOF-basierten 3D-Laserscanners stillgelegte Minen kartiert, um Informationen zur Stabilität des darüber liegenden Terrains zu gewinnen. Bei den Wettbewerben der DARPA Challenge traten zahlreiche internationale Teams mit PKW-basierten, autonomen Fahrzeugen an, die eine Vielzahl von Sensoren nutzten, vor allem

Laserscanner, stereometrische Verfahren und GPS-Empfänger (Stiller u. a. 2008; Berger und Rumpe 2008).

1.2.2 Entfernungssensoren

Zur automatisierten Entfernungsmessung existieren zahlreiche Verfahren, deren Grundprinzipien, typische Anwendungsbereiche und Einschränkungen hier kurz zitiert werden sollen. Die wohl wichtigste Unterscheidung ist die zwischen aktiven Sensorsystemen, die EM-Signale oder Schallwellen ausstrahlen, und passiven Systemen (Beraldin u. a., 1995; Trucco und Verri, 1998). Bei den passiven, kamerabasierten Systemen ist Stereo-Vision das wohl wichtigste Verfahren; zum “state of the art” gehört dabei z.Zt. das Verfahren “Semi-Global Matching” nach Hirschmüller (2008). Ein weiteres wichtiges passives Sensorprinzip ist “shape from motion” bzw. “optical flow”. Der Vorteil passiver Verfahren ist, dass 3D-Informationen ohne ausstrahlende Komponenten gewonnen werden; auch mechanische Aktoren sind meist nicht erforderlich. Jedoch haben diese Verfahren einen entscheidenden Nachteil. Um 3D-Informationen aus zweidimensionalen Bildern zu gewinnen, müssen bei passiven Verfahren Korrespondenzen zwischen Bildpunkten zeitlich oder räumlich divergenter Bilder hergestellt werden. Dies erfordert ein Mindestmaß an Features bzw. Textur im Bild: “In typical outdoor scenes, it is possible to achieve very dense stereo results” (Konolige u. a., 2006); bei unstrukturierten, einfarbigen Flächen, wie sie in Innenräumen typisch sind, fallen die Ergebnisse meist weniger gut aus. Die “Einstiegsklasse” der aktiven Sensorik bilden Sonar- und Infrarotsensoren. Diese sind verbreitet im Low Cost- und Mikrobereich; Reichweite und Genauigkeit solcher Sensoren sind oft mangelhaft. Demgegenüber bieten Laserscanner nach dem TOF-Prinzip (“time of flight”, Phasenverschiebung eines reflektierten Punktlasers; auch “LIDAR”, Light Detection and Ranging) sehr präzise Ergebnisse. Professionelle Geräte sind in aller Regel teuer und haben ein hohes Gewicht. Für einen 3D-Scan erfordern sie zwei genau zu kalibrierenden Achsen und damit einen hohen feinmechanischen und optischen Aufwand (vgl. Pito 1995).

TOF-basierte 2D-Laserscanner sind einfacher aufgebaut und preiswerter als 3D-Geräte. Sie sind auch bei autonomen Fahrzeugen bzw. industriellen Transportsystemen verbreitet. Jedoch erfassen diese Systeme keine Hindernisse außerhalb der Scanebene, was ihren Nutzen in unbekannter Umgebung einschränkt. Es ist möglich, zur Gewinnung von 3D-Daten einen solchen Sensor als ganzes um seine Querachse zu drehen, wie es z.B. bei Holz, Lörken und Surmann (2008) verwendet wird; dies führt jedoch zu einem hohen Gesamtgewicht; selbst kompakte 2D-TOF-Systeme wiegen ein Kilogramm oder mehr (s. SICK AG 2010) und machen entsprechend aufwendige Aktuatoren erforderlich.

Beim Lichtschnittverfahren wird eine Lichtebene über das zu scannende Objekt geführt. In der häufigsten Implementierungsweise wird diese Ebene durch rotierende Spiegel, zylindrische Linsen oder Spaltoptiken aus einem Laserstrahl erzeugt. Betrachtet man den Schnitt von gescanntem Objekt und Lichtebene aus einer seitlich versetzten Perspektive, wird ein Höhenprofil sichtbar, das man mit einer Kamera aufzeichnen kann. Aus den Bildkoordinaten des Höhenprofils, dem Abbildungsmodell der Kamera und der Ausrichtung von Laser und Kamera lassen sich 3D-Koordinaten des Höhenprofils triangulieren. Dieses Verfahren wurde bereits vor der Verbreitung digitaler Kameras angewendet (Davis und Chen, 2001; Chen und Kak, 1987) und für zahlreiche Spezialanwendungen angepasst (z.B. Blais 2004). Eine neuere Entwicklung dabei ist, dieses Prinzip mit einem Minimum an Hardware nutzbar zu machen. Insbesondere wird bei Winkelbach u. a. (2006) der Laser freihändig über das zu scannende Objekt geführt; hier ist allerdings hinter dem gescannten Objekt ein Kalibrierungsmuster aufzustellen, so dass dieses Verfahren für ein autonomes mobiles System nicht in Frage kommt. Allen Lichtschnittverfahren ist gemein, dass sie zur Bilderzeugung nicht auf optische Features angewiesen sind und darum gute Ergebnisse auch bei untexturierten Oberflächen liefern.

Eine Erweiterung des Lichtschnittverfahrens ist die Streifen- oder Musterprojektion; hierbei werden codierte Hell-Dunkel-Muster (Gray-Code bzw. Phase-Shift-Verfahren) auf das zu scannende Objekt projiziert und so simultan eine Vielzahl von

Lichtschnitten durchgeführt (Haberjahn, 2006; Rocchini u. a., 2001) Hierfür ist jedoch ein vergleichsweise komplexer und schwerer Videoprojektor erforderlich, so dass dieses Verfahren für ein leichtes autonomes System nicht in Frage kommt.

1.3 Ziel der Arbeit

Im skizzierten Feld autonomer Systeme und ihrer Sensoren besteht eine Lücke zwischen einerseits Systemen für den professionellen Einsatz, die mit Sensoren hoher Reichweite und Genauigkeit präzise Weltmodelle erstellen und mit diesen navigieren, und andererseits preisgünstigen Systemen für den Consumer-Bereich, die überwiegend noch auf eine einfache Kollisionsvermeidung und reaktives Verhalten beschränkt sind.

In dieser Arbeit soll daher ein autonomer, mobiler Roboter konstruiert, programmiert und erprobt werden, der sich in unbekannten Gebäuden orientiert, diese selbstständig erkundet und dabei eine Karte seiner Umgebung erzeugt. Dabei soll so weit wie möglich auf Hardwarekomponenten aus dem Consumerbereich und auf quellenoffene Software zurückgegriffen werden.

Als Sensor soll dabei ein auf dem Laserschnitt-Verfahren basierendes System entwickelt werden. Dieses Verfahren verspricht gute Ergebnisse unter den typischen Bedingungen in Innenräumen und lässt sich ohne besonders teure oder schwere Hardware implementieren.

Die Arbeit gliedert sich folgendermaßen: Kapitel zwei fasst die wichtigsten mathematischen Grundlagen zusammen, die in dieser Arbeit verwendet wurden.

Kapitel drei hat die entwickelten Verfahren, die Implementierung und Tests des entwickelten Sensors zum Inhalt. In Kapitel vier wird zunächst die Hardware- und Softwarearchitektur des autonomen Systems umrissen; anschließend werden das verwendete Weltmodell und die zur Navigation entwickelten Verfahren beschrieben, implementiert und getestet. Es folgen Schlusskapitel, Anhang und Anlagen.

2 Grundlagen

Kameramodell

Grundlage der projektiven Geometrie des entwickelten Sensors ist das Lochkameramodell nach Bradsky und Kaehler (2004, S. 370 ff.); siehe auch Luhmann (2003, S. 237 ff.). Zur Projektion eines 3D-Punktes Q auf den korrespondierenden Bildpunkt q fasst dieses folgende Größen in der Fundamentalmatrix M zusammen: den Kamerahauptpunkt c (die Projektion des Projektionszentrums C in die Bildebene) mit Bildkoordinaten c_x, c_y ; und je eine Brennweite f_x und f_y für beide Bildachsen. Durch die zweite Brennweite wird modelliert, dass die Abstände der Bildzellen des Kamerachips in Hoch- und Querrichtung verschieden sein können. Es gilt

$$q = MQ \text{ mit } q = \begin{pmatrix} x \\ y \\ w \end{pmatrix}, M = \begin{pmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix}, Q = \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}. \quad (2.1)$$

Durch die Verwendung einer Linsenoptik in der Kamera stellt das Lochkameramodell nur eine Approximation der Abbildung dar. Zur Korrektur des resultierenden Abbildungsfehlers wird das Lochkameramodell durch ein nichtlineares Verzerrungsmodell komplementiert. Dabei werden tangentielle und radiale Verzerrungseffekte je erster und zweiter Ordnung berücksichtigt. Es gilt

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = (1 + k_1 r^2 + k_2 r^4) \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} 2p_1 xy + p_2(r^2 + 2x^2) \\ p_1(r^2 + 2y^2) + 2p_2 xy \end{pmatrix} \quad (2.2)$$

Die Verzerrungsparameter werden zusammengefasst als $K = (p_1, p_2, k_1, k_2)$.

Photogrammetrische Grundbegriffe

Innere, relative, und äußere Orientierung sind Grundbegriffe der Photogrammetrie (Luhmann, 2003). Die innere Orientierung ist das Kameramodell; die äußere Orientierung beschreibt Lage und Ausrichtung eines räumlichen Bildkoordinatensystems in einem übergeordneten Weltkoordinatensystem. Die relative Orientierung beschreibt bei Stereobildverfahren die relative Ausrichtung zweier Stereokameras in einem gemeinsamen Koordinatensystem. In dieser Arbeit wird damit die relative Ausrichtung einer Kamera und eines Lasers als passiver bzw. aktiver Bestandteil eines Sensorsystems bezeichnet.

Hauptkomponentenanalyse

Die Hauptkomponentenanalyse (auch: Karhunen-Loève-Transformation; *Principal Component Analysis*, PCA) ist ein Verfahren der multivariaten Statistik. Sie entkorreliert m korrelierte Zufallssignale optimal, so dass diese mit minimalem quadratischen Rekonstruktionsfehler durch $n < m$ Signale ausgedrückt werden können. (Meffert und Hochmuth, 2004, S.91 ff.)

Hough-Transformation

Die Hough-Transformation ist ein Voting-basiertes Verfahren zur Detektion von abstrakten Features in diskreten Bildern. Zur Detektion von Geraden ist das Verfahren folgendes: Zum Urbild U wird ein Akkumulator-Bild A erstellt, das einen $\phi - \rho$ -Raum repräsentiert. ϕ und ρ stehen dabei für ideale Geraden $g_U : \rho = x \cos \phi + y \sin \phi$ in U . Jede diskrete Koordinate in U repräsentiert ein Büschel b von Geraden, die durch diesen Punkt verlaufen können; in A stellt sich dieses Büschel wiederum als Gerade g_A dar. Es werden nun für jedes Pixel in U , das über einem Schwellwert liegt, in A die Zellenwerte unter g_A inkrementiert. Führt man dies für das gesamte Urbild U aus, repräsentieren die lokalen Maxima in A die besten Annäherungen an eine Gerade in U ; sie stehen für

jene Geraden g_U , die die höchste Zahl von Pixeln als “Zeugen” aufweisen. Gegenüber der klassischen Hough-Transformation stellt die probabilistische Hough-Transformation eine Erweiterung dar, die als Resultat nicht Geraden, sondern Strecken im Urbild liefert. Für Details sei verwiesen auf Kälviäinen u. a. (1994).

Morphologische Operatoren

Die morphologischen Operatoren “Erosion” und “Dilatation” sind dem Bereich der Bildverarbeitung zuzurechnen. Bei der Dilatation wird jedes Pixel eines Bildes auf das Maximum, bei der Erosion auf das Minimum der Werte einer lokalen Umgebung gesetzt. Es gilt

$$\text{erode}(x, y) = \min_{\substack{-\delta_x \dots \delta_x \\ -\delta_y \dots \delta_y}} (x + \delta_x, y + \delta_y) \quad (2.3)$$

$$\text{dilate}(x, y) = \max_{\substack{-\delta_x \dots \delta_x \\ -\delta_y \dots \delta_y}} (x + \delta_x, y + \delta_y) \quad (2.4)$$

Canny-Filter

Der Canny-Filter extrahiert Kanten entlang der Farbwertgradienten eines Bildes; dabei werden zum einen Hysterese-Schwellwerte verwendet, die bezüglich einer lokalen Pixelumgebung adaptiv sind; zum anderen wird die Kantenextraktion mehrfach unter Annahme verschiedener S/N-Raten durchgeführt um die jeweils besten Ergebnisse zusammenzuführen. Für Details sei auf Canny (1986) verwiesen.

2.1 Kamerakalibrierung nach Zhang

Zhang (2000) entwickelte ein Verfahren, dass die Kalibrierung einer Kamera, also die Bestimmung von M und K , mit Hilfe einfacher, planarer Objekte ermöglicht. Dabei wird auf eine ebene Fläche ein schwarz-weißes Schachbrettmuster S mit quadratischen Feldern bekannter Kantenlänge s und bekannter Anzahl $m + 1, n + 1$ in horizontaler

bzw in vertikaler Richtung aufgebracht. Anschließend werden zunächst die Bildkoordinaten der Schachbrettecken $p_{m,n}$ mit der Methode nach Harris und Stephens (1988) in Sub-Pixel-Genauigkeit bestimmt. Mittels Singulärwertzerlegung lassen sich nun die optimalen Parameter einer Homographie zwischen den Bildkoordinaten und den Modellkoordinaten der Kreuzungspunkte bestimmen und daraus sowohl H als auch K ableiten.

Umgekehrt lässt sich das Verfahren bei bekannten Kameraparametern H und K dazu verwenden, um die Lage des Schachbrettmusters S im Raum zu bestimmen, in der OpenCV-Implementierung ausgedrückt als Translationsvektor P_0 vom Kamerahauptpunkt C zur Schachbrettecke $p_{0,0}$ und als Rotationsmatrix R relativ zur optischen Achse der Kamera. Die projektive Geometrie der Abbildung, die bei Transformationen von Bild- in Weltkoordinaten nur in Abhängigkeit von einem Skalierfaktor bestimmt werden kann, wird durch die bekannte Kantenlänge s der Schachbrettfelder eindeutig; s ist die Einheit des Kamerakoordinatensystems, in der P_0 und R zurückgegeben werden.

OpenCV

Eine wesentliche Grundlage für die im Rahmen dieser Arbeit erstellten praktischen Implementierungen ist die Programmbibliothek OpenCV (OpenSource Computer Vision; Bradsky und Kaehler 2004); sie implementiert zahlreiche grundlegende Methoden der Bildverarbeitung, darunter sämtliche oben genannten Verfahren.

3 Umfelderfassung

In diesem Kapitel wird die Funktionsweise des entwickelten Lichtschnitt-Sensors erläutert. Auf eine allgemeine Beschreibung des Funktionsprinzips folgt eine Beschreibung der zugrundeliegenden Geometrie und des entwickelten Kalibrierungsverfahrens. Anschließend werden Probleme der Bildverarbeitung und der Synchronisation behandelt, die für eine korrekte Abbildung zu lösen sind. Abgeschlossen wird das Kapitel mit einer Zusammenfassung und Bewertung der Ergebnisse. Zur Beschreibung eines Entfernungssensors sind eine Vielzahl von Parametern zu bestimmen, vgl. Beraldin u. a. (1995) und Besl (1988). Es seien an dieser Stelle die Ergebnisse der Entwicklung vorweggenommen und folgende Parameter des Sensors genannt:

- Bildgröße: $230 \cdot 320 = 73600$ 3D-Punkte; in höherer Auflösung (hier nicht verwendet) $230 \cdot 640 = 147200$ 3D-Punkte.
- Gesichtsfeld: $35.4^\circ \times 230 \cdot 0.9^\circ = 207^\circ$, seitlich begrenzt durch Anordnung des Sensors vor dem Fahrzeug
- Mindestentfernung ("Standoff distance"): 14.9 cm
- Aufnahmezeit eines Rundumscans bei Auflösung 320×240 : 12 s plus Verarbeitungszeit. Die experimentell ermittelte, maximale Bildrate der Kamera liegt bei der gewählten Auflösung bei 28.52 Hz. Im Scanbetrieb werden für die Aufnahme von 230 Einzelbildern ca. 12 Sekunden benötigt, das entspricht nur genau $\frac{2}{3}$ der Bildrate der Kamera; zur Erklärung siehe Abschnitt 3.3.3.

- Die seitliche Bildauflösung bestimmt sich durch die Schrittweite des Motors und die Entfernung d als $x = \sin(0.45^\circ) \cdot 2 \cdot d$. Sie ist damit bei der hier verwendeten Kameraauflösung bis ab einer Entfernung von ca. 45 cm größer als die exponentiell abnehmende Tiefenauflösung.
- Die Tiefenauflösung ist abhängig von der gewählten Kameraauflösung. Ohne Sub-Pixel-Interpolation ergeben sich für die Bildzeile des Kamerahauptpunkts folgende Werte:

×	320 × 240	640 × 480
15 cm	< 1 mm	< 0.5 mm
50 cm	< 1 cm	< 0.5 cm
1 m	< 4 cm	< 2 cm
2 m	< 15 cm	< 7 cm
3 m	< 35 cm	< 17 cm
4 m	< 75 cm	ca. 30 cm
8 m	ca. 2-3 m	ca. 1 m

3.1 Funktionsprinzip

Zur Umfelderkennung wird eine ungefähr orthogonal zum Boden stehende Lichtfläche, die durch einen Laser mit Zylinderoptik erzeugt wird, um ihre Hochachse rotiert. Außerhalb der Ebene L , in der die Lichtfläche liegt, befindet sich ein CMOS-Kameramodul. Es ist in seiner Position zum Laser durch eine Halterung fixiert; dieser Verbund wird von einem Schrittmotor gedreht. Die Kamera nimmt nach jedem Motorschritt ein Bild des Laserprofils auf. Vor Einsatz des Sensors wird, wie nachfolgend beschrieben, eine diskrete Funktion $LUT : (x, y) \rightarrow (X, Y, Z)$ in Form einer Tabelle (“Lookup-Table”) erstellt, die jedem Bildpunkt, in dem das Laserprofil festgestellt wird, eineindeutig eine Raumkoordinate relativ zum Sensorverbund zuordnet. In diese Tabelle geht die gesamte Geometrie von Kamera und Laser ein, einschließlich der Kameraverzerrung, so dass die endgültige Weltkoordinate nur noch

von der Rotation Φ des Schrittmotors abhängig ist. So kann ein dreidimensionales Panorama erstellt werden, indem der Sensor einmal durch seinen Aktionsbereich gedreht wird, in jedem Bild jede detektierte Bildkoordinate des Lasers in der Lookuptable ausgelesen wird und die resultierenden Punkte in Abhängigkeit von Φ um die Hochachse MA des Motors rotiert werden.

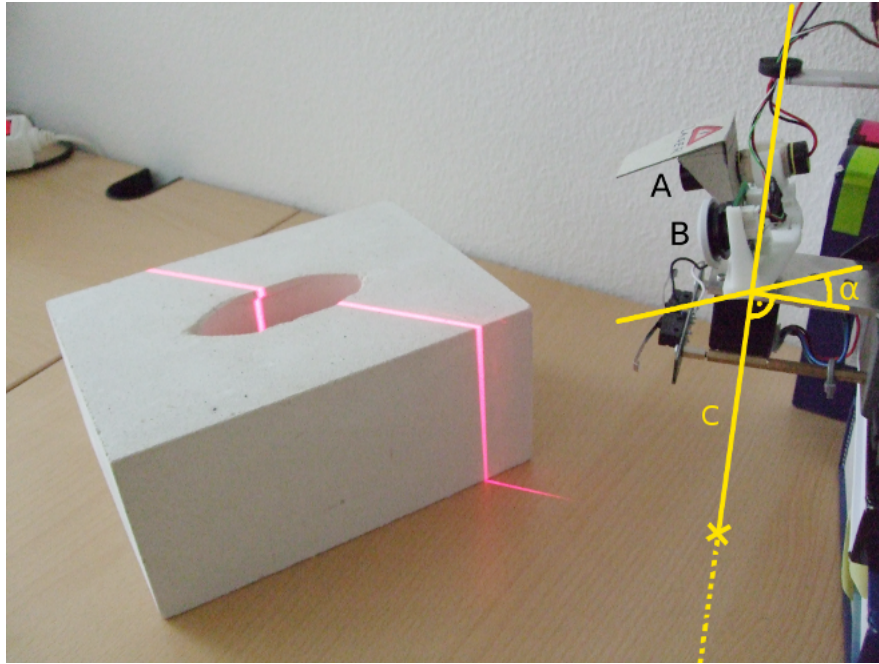


Abbildung 3.1: Der Sensor im Betrieb. A Laser, B Kamera, C angedeutete Lage der Rotationsachse mit Fußpunkt auf der Bodenebene. Sensor ist um $\alpha = 15^\circ$ in der Querachse geneigt.

3.2 Kalibrierungsverfahren

Chen und Kak (1987) stellen ein Verfahren zur Bestimmung der Geometrie zwischen Laser und Kamera vor, das z.B. auch bei Forest u. a. (2004) verwendet wird; dieses erfordert jedoch ein genau vermessenes, dreidimensionales Kalibrierungsobjekt. Demgegenüber wird hier, basierend auf der Kamerakalibrierung nach Zhang (2000), ein

Verfahren entwickelt, das mit einem planaren, bedruckten Objekt funktioniert, das deutlich einfacher herzustellen ist.

Während bei den meisten Verfahren zur Lasertriangulation die Kamera fest montiert ist und der Laser bewegt wird, erlaubt die Verwendung kleiner, leichter, auf einem einzigen Chip untergebrachter Kameras, wie sie z.B. in Webcams verwendet werden, den Laser und die Kamera zueinander zu fixieren und zum Abtasten der Umgebung diesen Verbund als ganzes zu drehen. Der Vorteil bei dieser Anordnung liegt vor allem darin, dass die Kalibrierung weniger komplex ausfällt, da sie nur für eine einzige Konfiguration ausgeführt werden muss. So kann jedem Pixel, in dem das Laserprofil festgestellt wird, eine feste 3D-Weltkoordinate zugeordnet werden, die nur noch von der Drehung des Gesamtverbunds abhängig ist.

Ein weiterer Vorteil dieser Anordnung ist ein nur von der drehbaren Lagerung abhängiges Gesichtsfeld von theoretisch bis zu 360° . Bei anderen Anordnungen wird das Gesichtsfeld des Sensors oft durch das Gesichtsfeld der Kamera beschränkt; um letzteres dennoch optimal zu nutzen werden aufwendige optische Maßnahmen notwendig (Stichwort: Scheimpflug-Bedingung, Besl 1988; Blais 2004).

Bei der Kalibrierung des Scanner sind also gesucht: die innere Orientierung der Kamera; die relative Orientierung von Kamera und Laser; und die äußere Orientierung des Kamerakoordinatensystems zu einem allgemeinen Sensorkoordinatensystem (Luhmann, 2003).

3.2.1 Innere Orientierung der Kamera

Zur Kalibrierung der Kamera, also der Bestimmung ihrer inneren Orientierung, wird das im Grundlagenkapitel vorgestellte Verfahren nach Zhang (2000) verwendet. In der Programmbibliothek OpenCV wird zur Bestimmung der Bildpunkte die Methode nach Harris und Stephens (1988) verwendet. Diese wird auch im nächsten Schritt dieser Arbeit verwendet werden, um die relative Orientierung von Kamera und Laser zu bestimmen. Zur Bestimmung der inneren Orientierung ist diese Methode auch verwendbar; praktisch wurden Passpunkte mit höherer Genauigkeit bestimmt, indem

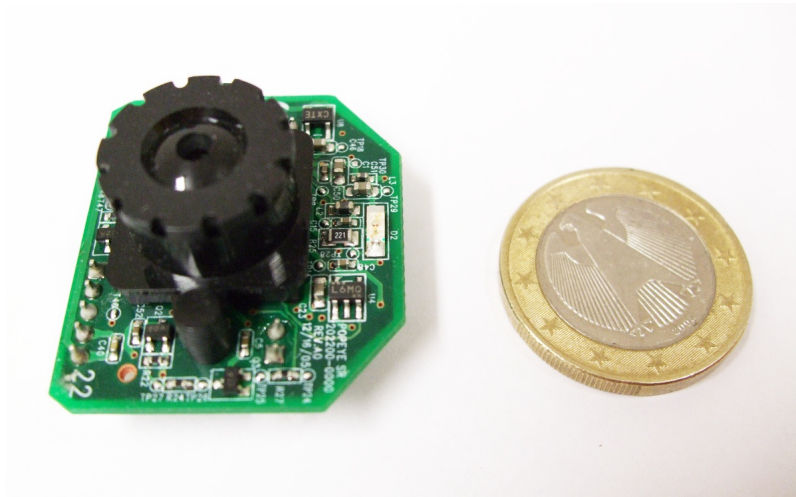


Abbildung 3.2: verwendetes CMOS-Kameramodul

kreisförmige Targets auf einer Kalibrierungswand mit Hilfe der Software Australis bestimmt wurden. Vor der Bestimmung der inneren Orientierung, die nicht mit Australis, sondern mit der in OpenCV implementierten Methode nach Zhang erfolgte, sind die Koordinatensysteme der Passpunkte umzurechnen; dazu wurde ein Hilfsprogramm nach Lemke (2009) verwendet. Resultat des Verfahrens sind Fundamentalmatrix H und Verzerrungsparameter K der Sensorkamera.

3.2.2 Relative Orientierung von Kamera und Laser

Im folgenden wird die Bestimmung der relativen Orientierung zwischen Kamera und Laser beschrieben. Diese ist nur einmalig zu bestimmen, da Kamera und Laser fest zueinander orientiert sind. Wegen der relativ kleinen Basis zwischen Laser und Kamera führen dabei schon kleine Ungenauigkeiten in der Ausrichtung zu großen Abbildungsfehlern. Anstatt also zu versuchen, eine präzise Positionierung des Lasers vorzunehmen und daraus die Scannergeometrie zu berechnen, wird die Laserebene nur ungefähr orthogonal zum Boden ausgerichtet und ihre tatsächliche Ausrichtung aus Kalibrierungsbildern bestimmt.

Grundlage ist das bereits zur Bestimmung der inneren Orientierung verwendete Verfahren nach Zhang (2000). Als Kalibrierungsobjekt dienen planare, schwarz-weiße Schachbrettmuster mit quadratischen Feldern bekannter Kantenlänge s und bekannter Anzahl $m + 1, n + 1$ in horizontaler bzw. vertikaler Richtung. Außerdem wird in jeder Aufnahme das Schachbrett so positioniert, dass die Laserebene das Schachbrett schneidet.

Die eigentlichen Kalibrierungspunkte sind die Kreuzungspunkte $p_{m,n}$ der quadratischen Felder. Bestimmt man aus dem Kalibrierungsbild i die Lage des Schachbretts S_i im Raum, und die Bildkoordinaten l_i der auf dem Schachbrett sichtbaren Laserlinie L_i , lässt sich die Lage von L_i im Raum als Schnitt der Laserebene L mit dem Schachbrett S_i bestimmen. Aus mindestens zwei verschiedenen solchen Bildern lassen sich damit zwei verschiedene, in L liegende Geraden L_i bestimmen, und damit genug Informationen, um die Ebene L zu ermitteln (vgl. Abb. 3.3).

Um die Lage der durch das Schachbrett definierten Ebene zu bestimmen, wird die in OpenCV implementierte Umkehrung des Verfahrens nach Zhang genutzt; siehe Kapitel “Grundlagen”.

Als Ergebnis erhält man die Lage der Schachbrettebene S_i im Raum, ausgedrückt als Translationsvektor P_0 vom Projektionszentrum C zur Schachbrettecke $p_{0,0}$ und als Rotationsmatrix R relativ zur optischen Achse der Kamera. Als Norm dient die bekannte Kantenlänge s der Schachbrettfelder. Auf diese Weise wird ein Maßstabsfaktor in die perspektivische Transformation eingebracht, die an sich hinsichtlich des Skalierfaktors invariant ist.

Um die Lage der Laserebene zu bestimmen, werden die Kalibrierungsbilder so eingerichtet, dass in jedem Bild das Schachbrettmuster vollständig zu sehen und der Sensor so ausgerichtet ist, dass die Laserebene das Schachbrettmuster schneidet. Dabei werden für jede Ausrichtung zwei Bilder aufgenommen, ein Bild B_1 mit ein- und ein Bild B_2 mit ausgeschaltetem Laser. Um in einem ersten Schritt die den Laserschnitt repräsentierenden Pixel bestmöglich zu extrahieren, werden B_1 und B_2 in Grauwertbilder umgewandelt und das Bild B als absolute Differenz aus B_1 und B_2

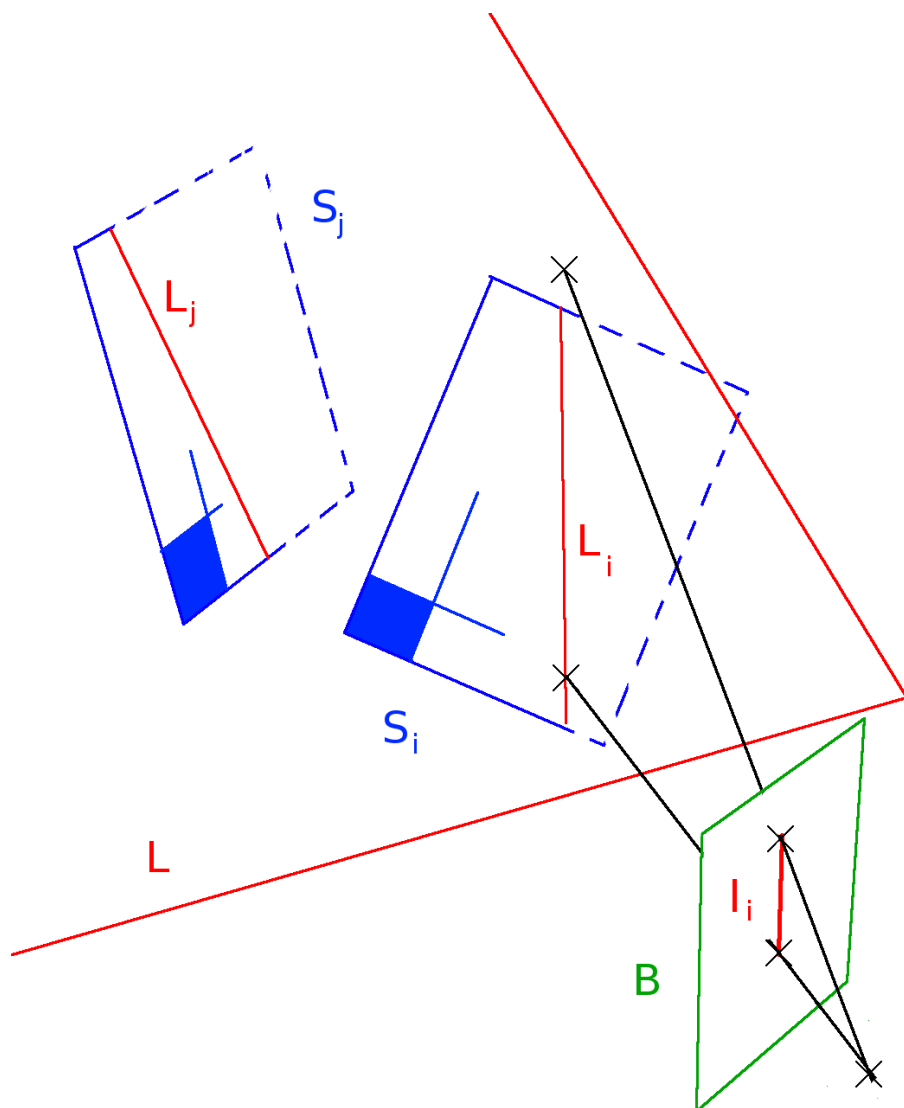


Abbildung 3.3: Kalibrierung der relativen Orientierung. Laserebene L ; Ebenen der Kalibrierungsmuster S_i, S_j ; 3D-Schnittlinien L_i, L_j ; Projektion l_i von L_i in die Bildebene B .

bestimmt. Der Zeitaufwand der doppelten Aufnahme und insbesondere die Entstehung von Fehlern im Differenzbild bei jeder relativen Bewegung von Kamera und Szene stellen in der kontrollierten Umgebung der Kalibrierungsphase kein Problem dar. Für die Betriebsphase des autonomen Systems ist eine andere Methode zur Extraktion des Laserbildes zu verwenden.

Das Differenzbild B wird entzerrt und mit einem heuristisch bestimmten Schwellwert in ein Binärbild umgewandelt. Auf diesem wird ein rechteckiges Fenster (“Region of Interest, ROI”) so gewählt, dass es durch die äußersten Schachbrettpunkte bestimmt wird (Abb. 3.7). Innerhalb dieses Fensters wird nun die Laserlinie mittels einer nicht-probabilistischen Houghtransformation bestimmt. Es ist günstig, einen relativ hohen Schwellwert für die Binarisierung zu wählen, da die Laserlinie aufgrund von Überbelichtung im Originalbild oft mehrere Pixel breit erscheint; mit einem hohen Schwellwert bleiben von der Linie nur noch die hellsten, in der Regel zentralen Pixel übrig; die entstehenden Diskontinuitäten beeinträchtigen nicht den “best guess” der Hough-Funktion.

Nach dem Lochkameramodell aus Gleichung 2.1 bestimmt jeder Bildpunkt p eine Gerade g_p im Raum:

$$g_p = \lambda \cdot \vec{a} \text{ mit } \vec{a} = \begin{pmatrix} \frac{x-c_x}{f_x} \\ \frac{y-c_y}{f_y} \\ 1 \end{pmatrix}. \quad (3.1)$$

Bringt man die Schachbrettebene S_i in Normalenform $\vec{r} \cdot \vec{n} = d$ und setzt g_p für \vec{r} ein, erhält man λ , und damit für jedes p den korrespondierenden, auf S_i liegenden Raumpunkt $P = (X, Y, Z)^T$ als Schnittpunkt $S_i \cap g_p$ mit

$$P = \frac{d}{\vec{a} \cdot \vec{n}} \cdot \vec{a} \quad (3.2)$$

Setzt man auf der in Bildkoordinaten vorliegenden Linie L_i zwei Punkte l_i^1, l_i^2 beliebig fest, repräsentieren diese nach Gleichung 3.1 zwei Lichtstrahlen $s_{i,1}, s_{i,2}$. Diese schneidet man mit S_i ; die resultierenden Punkte, die nicht notwendigerweise innerhalb

des Schachbrettmusters liegen, definieren den räumlichen Laserschnitt L_i auf dem Kalibrierungsobjekt.

Führt man die bis hierhin beschriebene Prozedur für zwei Aufnahmen durch, bei denen die räumliche Orientierung der Schachbretter bezogen auf die Kamera verschieden ist, kann man bereits durch drei der vier gewonnenen Punkte eine Gleichung für die Laserebene L bestimmen. Um numerisch günstige, d.h. vor allem hinreichend verschiedene L_i zu erhalten, empfiehlt sich die Verwendung von mehr als zwei Bildern; als geeignet hat sich eine Kombination zweier Setups erwiesen, zwischen denen das Schachbrett um ca. 90° um die Normale der Laserebene gedreht wird (Abb. 3.3). In beiden Setups erfolgt dann eine Serienaufnahme, bei der der Kamerakopf so gedreht wird, dass die Laserlinie je einmal über das gesamte Schachbrett geführt wird.

Fasst man die Menge der so gewonnenen n Vektoren, die bis auf einen unbekannten Fehler in der Ebene L liegen, als Zufallssignale auf, lässt sich das Problem der Suche nach einer optimalen Ebenengleichung mit Methoden der multivariaten Statistik lösen, genauer: als Hauptkomponentenanalyse (auch Karhunen-Loève-Transformation). Hierbei werden die L_i als korrelierte Zufallssignale aufgefasst, die man optimal entkorreliert und mit minimalem quadratischen Rekonstruktionsfehler durch ein einziges Signal ausdrücken möchte.

Die Laserebene L geht durch den Mittelpunkt \bar{l} aller gefundenen Punkte; Der mit dem größten Eigenwert korrespondierende, normalisierte Eigenvektor $\frac{EV_i}{|EV_i|} = \vec{n}$ ist ihr Normalenvektor.

Für den Abstand von L zum Ursprung des Kamerakoordinatensystems gilt damit $d = -\bar{l}_x \cdot \vec{n}$. Damit ist die Gleichung der Laserebene vollständig bestimmt.

Nun berechnet man für jeden ganzzahligen Bildpunkt p unter Berücksichtigung der Verzerrungsparameter K den korrespondierenden entzerrten Punkt p' auf der Bildebene und schneidet die Gerade durch c und p' nach dem bekannten Verfahren mit der Laserebene L ; so erhält man eine Raumkoordinate für jedes Pixel, in dem die Abbildung des Laserprofils detektiert wird.

3.2.3 Äußere Orientierung des Sensors

Die äußere Orientierung des Sensors ist die Transformation vom Kamerakoordinatensystem XYZ_K in ein Fahrzeug-zentriertes Koordinatensystem XYZ_F in Abhängigkeit vom Stellwinkel des Sensors, Φ (Abb. 3.2.3). $\Phi = 0$ wird dabei so festgelegt, dass die Ausrichtung des Lasers parallel zur Fahrtrichtung ist. Als Nullpunkt von XYZ_F wird der Punkt 0_F festgelegt, an dem die Achse des Schrittmotors M die ideale Bodenebene B schneidet, auf der das Fahrzeug steht. Die Achse des Schrittmotors als Hochachse zu setzen hat den Vorteil, die 3D-Koordinaten der einzelnen Weltschnitte abhängig vom Stellwinkel des Motors Φ leicht nach KYZ_K überführen zu können. Die Entfernungsachse Z_F des Fahrzeugkoordinatensystems liegt in der Bodenebene B und zeigt gegen die Fahrtrichtung. Die Hochachse Y_F zeigt nach oben, die seitliche Achse X_F nach links.

Um die gesuchte Transformation zu bestimmen, wird auf die Konstruktionsdaten des Fahrzeugs zurückgegriffen. Mit diesen wird eine kinematische Kette von Rotationen und Translationen so bestimmt, dass sie jeden Punkt P_K vom Kamerakoordinatensystem in seine Entsprechung P_F im Fahrzeug-Koordinatensystem XYZ_F überführt.

Dabei sind folgende Koordinatentransformationen zu berücksichtigen: Zuerst führt vom Nullpunkt des Kamerakoordinatensystems K_0 eine Translation $t_f = (0, 0, -f)$ auf das Zentrum des CMOS-Chips. Als nächstes bringt uns eine Translation t_p vom Zentrum des Kamerachips zu einem Punkt des Kamerahalters, der mit der äußeren, unteren Ecke der CMOS-Platine zusammenfällt. Es folgt eine Rotation von $\beta = 5^\circ$ um die Hochachse des Halters, also eine Korrektur des Winkels, mit dem die Kamera zur Laserebene hin gedreht ist, und eine Rotation von $\alpha = 15^\circ$ um die Querachse, die die Neigung des Kamerahalters zur Bodenebene korrigiert. Wir haben nun zur Transformation ins Fahrzeugkoordinatensystem nur noch Translationen auszuführen: eine Translation t_{r1} , die uns von der Ecke der CMOS-Platine auf eine äußere Ecke e des Kamerahalters bringt; und eine letzte Translation t_{r2} bringt zum Nullpunkt 0_F des Fahrzeugkoordinatensystems XYZ_F . Diese Transformation bestimmt man am besten als kinematische Kette homogener Koordinatentransformationen.

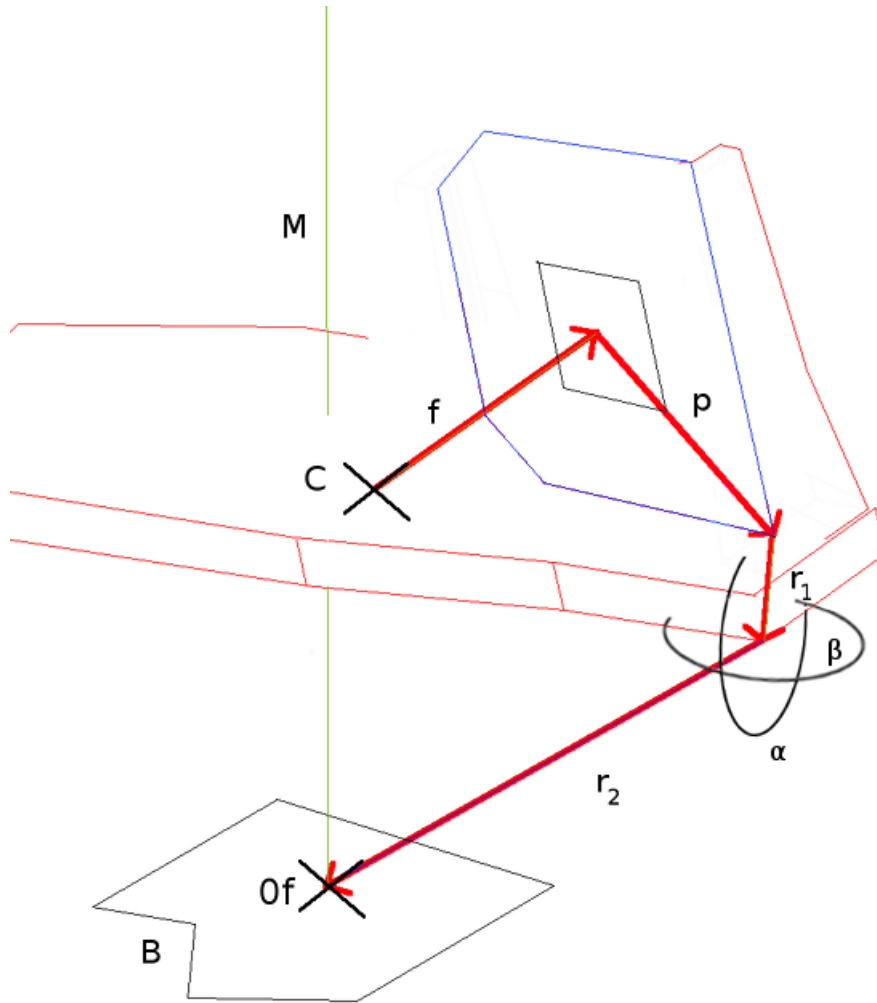


Abbildung 3.4: Äußere Orientierung des Sensors. C Projektionszentrum; M Motordrehachse; 0_f Nullpunkt des fahrzeugbezogenen Koordinatensystems in der idealen Bodenebene B . Verschiebungsvektoren f (Brennweite), p (von der CMOS-Platine auf den Kamerahalter), r_1 auf einen äußeren Punkt des Kamerahalters. Rotation um $\alpha = 15^\circ$ um die Quer- und $\beta = 5^\circ$ um die Hochachse gemäß Konstruktion des Halters. Translation r_2 nach 0_f gemäß Vermessung.

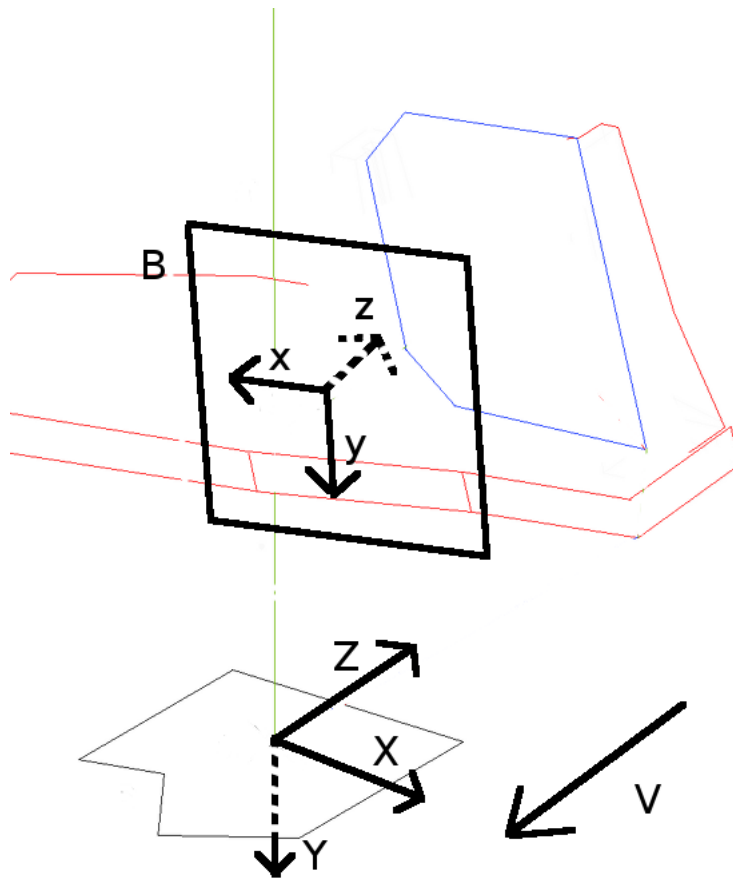


Abbildung 3.5: Koordinatensysteme. Kamerakoordinatensystem xyz in Bildebene B und Hauptpunkt c (nicht eingezeichnet); Fahrzeugzentriertes Koordinatensystem XYZ im Fußpunkt der Schrittmotorachse; Vektor V der Vorwärtsfahrt.

Es wird t_R am Aufbau vermessen, da für das Fahrgestell selbst keine Maßzeichnungen zur Verfügung stehen. Dabei wird angenommen, dass die Grundplatte des Fahrzeugs (vgl. Abb 4.2) parallel zu B und die Motorachse M orthogonal zu beiden ist. Die Lage des Sensorgitters auf der Kameraplatine, also t_H , lässt sich nur abschätzen, da Maßzeichnungen nicht verfügbar sind. Die übrigen Parameter werden aus der inneren Orientierung der Kamera bzw. den CAD-Plänen des Kamerahalters entnommen. Den Übergang vom Koordinatensystem eines einzelnen Laserschnitts in das eines gesamten Scans ist damit der einzige verbleibende Abbildungsschritt, der nicht in die LUT integriert wird. Da der Ursprung des Sensorkoordinatensystems in den Fußpunkt der Motorachse gelegt wurde, ist nur eine Rotation abhängig von Φ zu berechnen. Für eine komplette Serie S von Schnitten s mit Punkten P über den gesamten Winkel $FOV = \text{Schrittzahl} \cdot 0.9^\circ$, die zu gleichen Teilen beidseitig der Fahrtrichtung liegt, erfolgt die Zuordnung zu Punkten P' des 3D-Panoramas nach Algorithmus 1:

Algorithmus 1 2D-Schnitte \rightarrow 3D-Raum

```

 $\Phi \leftarrow -\frac{1}{2}(FOV - 180^\circ)$ 
for  $s \in S$  do
    for  $P \in s$  do
         $P'.x \leftarrow P.y \cdot \sin(\Phi) - P.x \cdot \cos(\Phi)$ 
         $P'.y \leftarrow P.x \cdot \sin(\Phi) + P.y \cdot \cos(\Phi)$ 
    end for
     $\Phi \leftarrow \Phi + 0.9^\circ$ 
end for

```

3.3 Implementierung

3.3.1 Mechanische Konfiguration

Zur Anordnung der Komponenten (vgl. Bild 3.1): Der Laser ist annähernd orthogonal zum Boden angeordnet. Er wird im Drehzentrum des Sensorverbunds platziert. Die Kamera wird seitlich versetzt dazu angebracht. Sie ist um 5° zur Laserebene gedreht, so

dass die Horizontlinie der Laserebene im Blickfeld liegt; die Reichweite des Sensors ist damit von der Kamerageometrie her theoretisch unbegrenzt, wenn man die diskrete Auflösung der Kamera und die resultierende begrenzte Tiefenauflösung vernachlässigt. Außerdem wird die Kamera um 15° zum Boden geneigt, um die Mindestsichtweite ("Standoff distance") im Nahbereich zu verbessern. Die Mindestreichweite ist durch diese Drehung auf 14.9 cm festgelegt; dies bezieht sich auf eine ebene Standfläche des Fahrzeugs. Außerdem ist die Kamera um 90° um ihre Längsachse gedreht. So kann das größere Gesichtsfeld in der Kamera-Querachse genutzt werden; das horizontale Gesichtsfeld wird ohnehin allein durch die Drehung des Kameraverbands bestimmt. Für diese Drehung wird ein Schrittmotor verwendet, mit dem ohne elektrisch-mechanische Kalibrierung genaue Winkel angesteuert werden können. Für eine initiale Positionsbestimmung wird ein einfacher Endschalter verwendet. Der zweiteilige Rotor, der die Verbindung zwischen Schrittmotor, Laser und Kamera herstellt, wurde mit Hilfe von CAD-Software entworfen und im Rapid-Prototyping-Verfahren mit einem 3D-Drucker hergestellt.

3.3.2 Automatisierte Kalibrierung

Vor der Kalibrierung der Geometrie müssen sowohl die Kamera als auch der Laser manuell fokussiert werden. Zur Fokussierung der Kamera wird ein Hilfsprogramm verwendet, das Bilder aus einem Videostream liest und den Kontrast des aktuellen Bildes kontinuierlich als Summe der Farbwertdifferenzen benachbarter Pixel ausgibt. Die Fokussierung erfolgt als Kontrastmaximierung mit Blick auf eine weit entfernte Szene.

Vor der Fokussierung des Lasers stellt sich die Frage, auf welche Entfernung die Brennweite optimal kalibriert werden soll. Dazu wurde die scheinbare Dicke der Laserlinie experimentell auf verschieden weit entfernte Ziele optimal kalibriert und dann jeweils auf verschiedene Entfernungen gemessen. Dabei ergab sich, dass eine Fokussierung des Lasers auf 2 m Entfernung den besten Kompromiss darstellt (s. Anhang A).

Zum Aufzeichnen geeigneter Kalibrierungsbilder kann ein Hilfsprogramm auf dem ARP-117 verwendet werden, das Kamera und Laser entsprechend steuert und die Bildpaare mit und ohne Laserschnitt speichert. Dabei sollte darauf geachtet werden, dass verschiedene Ausrichtungen des Schachbrettmusters durchlaufen werden. Es ist die höchstmögliche Kameraauflösung zu wählen.

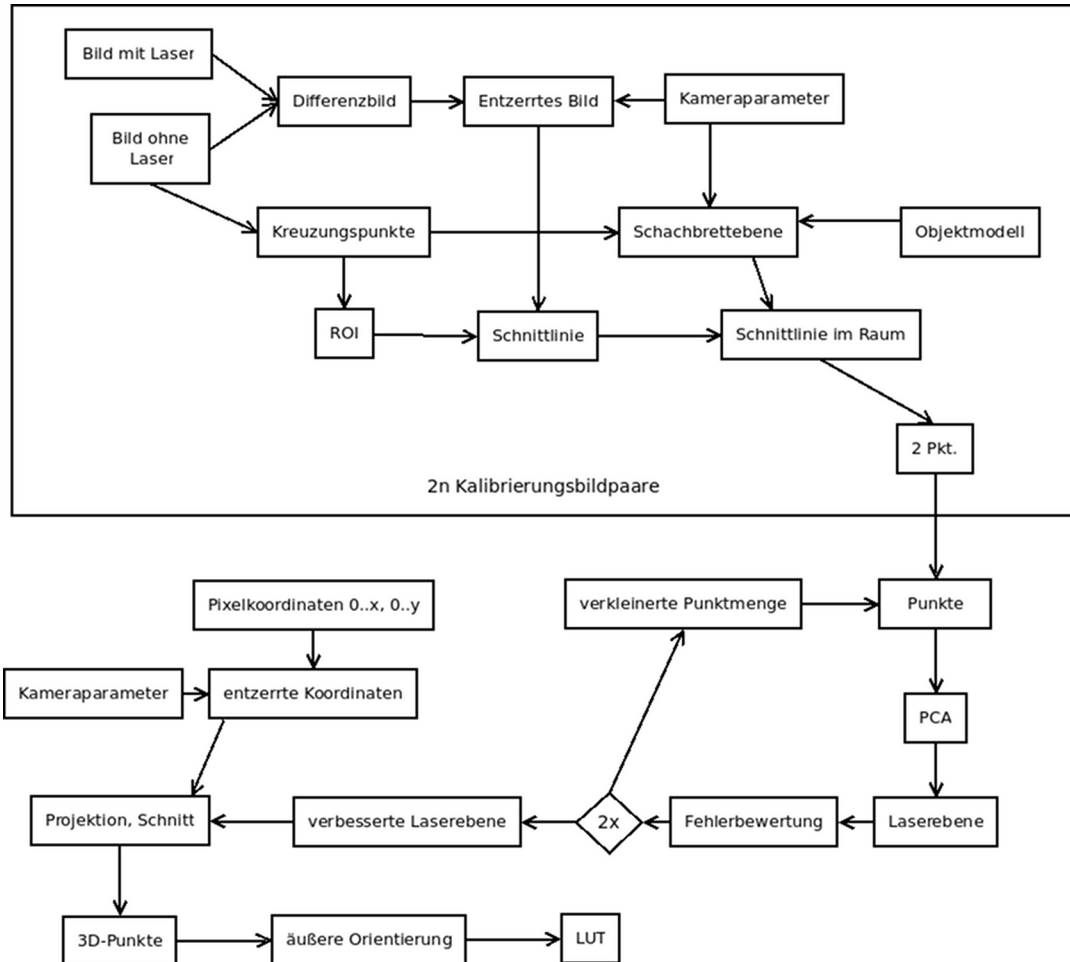


Abbildung 3.6: Implementierung der Kalibrierung im Programm **geometry**.

Das Kalibrierungsprogramm **geometry** implementiert die Kalibrierung auf der gewünschten Menge von Bildpaaren. Dabei werden ungeeignete Bilder automatisch verworfen. Es gelten folgende Kriterien: In jedem Bild muss ein Schachbrett mit der vorgegebenen Anzahl von Kreuzungspunkten erkannt werden; dies darf nicht zu nah am

Rand liegen; hier sind Fehlerkennungen möglich, der Mindestabstand vom Rand wurde heuristisch festgesetzt; und es muss eine Houghlinie erkannt werden, die einem bestimmten Schwellwert genügt. Nach Durchlaufen aller Bilder und der Extraktion der gesuchten, in der Laserebene liegenden 3D-Punkte wird diese per Hauptkomponentenanalyse bestimmt. Anhand des Resultats kann für jeden Punkt ein Fehlerwert bestimmt werden (Quadrat des Abstands zur Ebene L). Um “Ausreißer” zu eliminieren, wird die Hauptkomponentenanalyse dreimal durchlaufen; nach jedem Durchgang werden all jene Punkte verworfen, deren quadratische Fehlersumme über dem durchschnittlichen Fehler des Durchlaufs liegt.

3.3.3 Bilderfassung

Beim Scanvorgang wird der Sensorverbund einmal durch seinen gesamten Aktionsbereich von 230 Schritten zu 0.9° gedreht und für jede Stellung des Schrittmotors ein Bild aufgenommen. Jedes Bild wird zeilenweise (beachte Drehung der Kamera, Abb. 3.5) nach dem Laserpeak abgetastet und die korrespondierende 3D-Koordinate aus der LUT gelesen; so werden pro Laserschnitt bei der hier verwendeten Kameraauflösung von 320×240 Pixeln pro Bild 320 3D-Punkte gewonnen, die noch mit der Rotation Φ um die Kamera-Hochachse multipliziert werden müssen, um die endgültige Koordinate im 3D-Panorama zu erhalten (vgl. 1).

Der Kamera-Chip bietet keine Möglichkeit zur getriggerten Aufnahme zu einem angeforderten Zeitpunkt. Stattdessen werden im Video-Modus permanent Bilder aufgenommen, Kamera-intern ins JPEG-Format komprimiert und (mit einer Bildrate von 28.5 Hz bei der gewählten Auflösung) auf den USB-Bus geschickt. Im PC ist das jeweils aktuelle Bild an einer festen Speicheradresse (Framebuffer B) auszulesen. Um die Bewegung des Microcontroller-gesteuerten Schrittmotors präzise mit der Bildrate zu synchronisieren, wird jeweils ein Bild aus dem variablen Framebuffer an eine eigene Speicheradresse A kopiert. Nach Beenden der entsprechenden Funktion wird über den USB-UART-Wandler ein Befehl an den Mikrocontroller geschickt, um den Sensor einen Schritt weiter zu bewegen. Es ist nur statistisch vorauszusagen, ob zu diesem Zeitpunkt

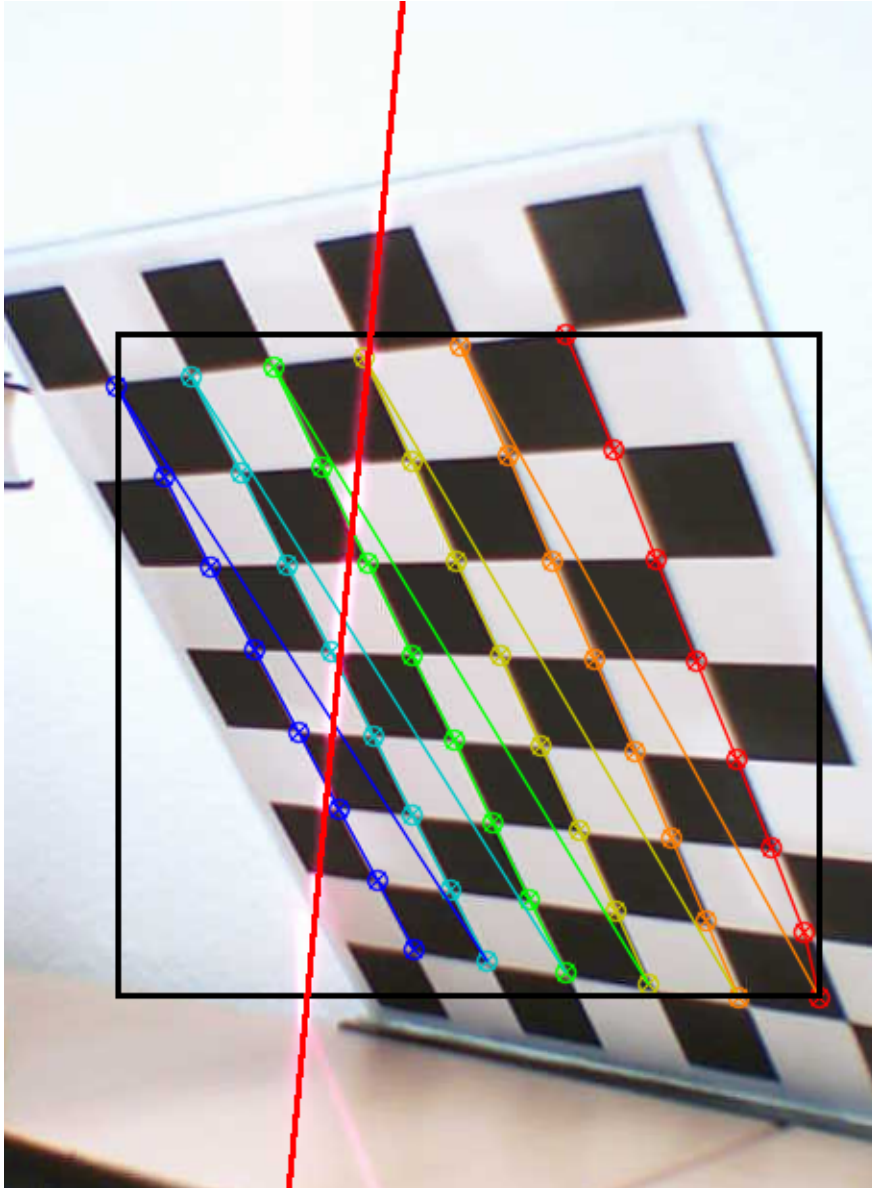


Abbildung 3.7: Analyse eines Einzelbildes im Kalibrierungsverfahren; Lage der Schachbrettpunkte, ROI aus deren Eckpunkten $p_{0,0}, p_{m-1,n-1}$, darin erkannt Laserprofil

bereits damit begonnen wurde, ein neues Bild in den Framebuffer zu schreiben. Wenn dies der Fall ist, enthält der Framebuffer kein korrektes Bild, sondern variable Teile von zwei verschiedenen Bildern. Die Systemfunktion zum Kopieren von B nach A wird dies erkennen, den fehlerhaften Inhalt verwerfen und blockierend erst das nächste vollständige Bild zurückgeben. Im Mittel wird so die Hälfte aller Bilder verworfen werden, eines von zwei Bildern also nur nach der doppelten Zeit geliefert werden. Die durchschnittliche Bildrate verringert sich damit um den Faktor $\frac{2}{3}$ zu 19.0 Hz; die Aufnahme von 230 Einzelbildern eines kompletten Scans dauert damit 12.1 Sekunden. Die verwendete Kamera hat keine Blende im eigentlichen Sinn; mehrere der relativ lichtschwachen Einzelbilder werden im Kamerachip noch vor der Übertragung per USB-Bus aufaddiert. Diese "Softwareblende" muss per Systemfunktion deaktiviert werden, um eine Bewegungsunschärfe aufeinanderfolgender Bilder zu vermeiden. Die Laserlinie ist dann mit bloßem Auge in einem Einzelbild zum Teil nicht mehr zu erkennen, der Farbwert ist aber noch hoch genug für eine Peakerkennung.

3.3.4 Bildverbesserung

Nachdem innerhalb der einzelnen Bilder keine Annahmen über Kontinuitäten der gescannten Objekte getroffen wurden, wird eine solche Kontinuität nun zwischen Folgebildern vorausgesetzt. Auf diese Weise eliminiert man einzelne falsch-positive Punkte. Die Einzelbilder werden dabei folgendermaßen gefiltert: Ein Bildpunkt p wird nur dann in den weiteren Schritten als Repräsentation eines 3D-Objektes angenommen, wenn entweder im vorausgehenden oder im nachfolgenden Bild in der gleichen Bildzeile ein Punkt auftaucht, dessen Spaltenabschnitt innerhalb eines heuristisch bestimmten Schwellwertes liegt. Da diese Filterung auf Pixelkoordinaten basiert, skaliert sie angemessen mit der quadratisch variablen Tiefenauflösung.

3.3.5 Peakdetektion

Aus jedem Einzelbild wird der Rotkanal extrahiert und zeilenweise nach der Laserlinie abgetastet. Da der Laser ungefähr orthogonal zum Boden orientiert ist, kann

Algorithmus 2 despeckle

$thresh \leftarrow 2$ (heuristisch bestimmt)

geg. Bilder $B_0 \dots B_n$ eines Rundumscans mit Zeilenmaxima $y_{m,0} \dots y_{m,x}$

for $a \leftarrow 1 \dots n$ **do**

for $b \leftarrow 0 \dots x$ **do**

if $(|B_{a,b} - B_{a,b-1}| > thresh)$

$\vee (|B_{a,b} - B_{a,b+1}| > thresh)$ **then**

$B_{a,b} \leftarrow 0$

end if

end for

end for

näherungsweise davon ausgegangen werden, dass die Linie jede Bildzeile genau einmal schneidet. Weil an dieser Stelle keine Annahmen über Formen in der realen Welt getroffen werden sollen, ist die Suche nach der Laserlinie auf zeilenweise Bildverarbeitungsmethoden beschränkt.

Da die Kamera über keine Hardware-Blende verfügt, ist sie empfindlich gegenüber Lichteinflüssen. Insbesondere bei einem Blick von einem dunkleren in einen von direktem Sonnenlicht beleuchteten Raum hat man eine starke Überbelichtung, was bei dessen Detektion auch dann ein Problem darstellt, wenn dieser eigentlich in einem dunkleren Bereich noch sichtbar ist - eine einfache zeilenweise Suche nach dem maximalen Helligkeitswert ist also in so einem Fall nicht ausreichend.

Es ist also eine Methode gesucht, mit der nicht größere, helle Flächen, sondern nur "kurze" Peaks als Laserlinie erkannt werden, also innerhalb einer Zeile: starke Anstiege von Helligkeitswerten gefolgt von einem starken Abfall. Dabei ist auch zu beachten, dass die Laserlinie auf kurze Entfernung mehrere Pixel dick erscheint, von denen viele den maximalen Helligkeitswert annehmen.

Die sicherste Detektion der Laserlinie im Bild erreicht man durch ein Differenzbild, wie es im Kalibrierungsverfahren verwendet wurde. Dies ist jedoch im Betrieb des

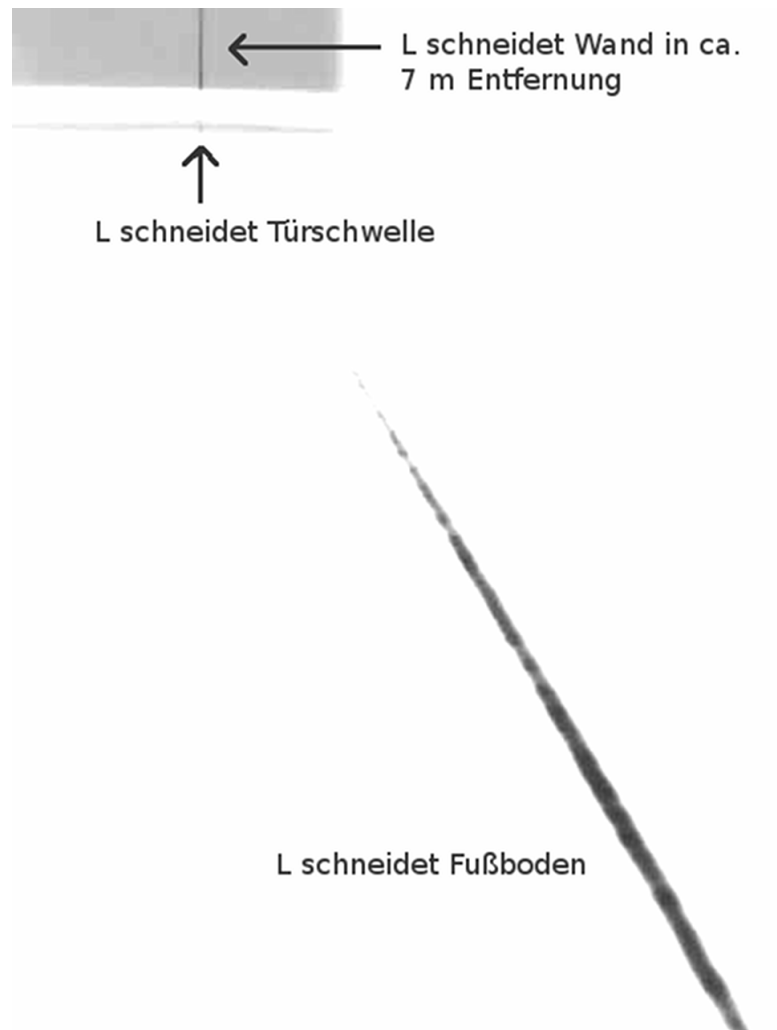


Abbildung 3.8: Scannerbild, hellkeitsinvertiert. Zu sehen sind der Schnitt der Laserebene L mit dem Fußboden und mit entfernten Hindernissen. An diesem Bild kann gezeigt werden: die geringere Reichweite der Bodenerkennung gegenüber der von senkrechten Hindernissen; Störungen durch spiegelnde oder helle Flächen (hier: Türschwelle aus Metall); und die hohe scheinbare Dicke der Laserlinie im Nahbereich.

autonomen Systems wegen möglicher Bewegungsstörungen und der Halbierung der Aufnahmegeschwindigkeit nicht praktikabel.

Die hier implementierte Lösung besteht im Prinzip aus einer Korrelation der Helligkeitsfunktion mit einem Dirac-Impuls. Um die Berechnung zu beschleunigen, die für je 320 Bildzeilen in 230 Bildern vorgenommen werden muss, kann dabei auf eine Normierung der Korrelation verzichtet werden; um die meist breiter als ein Pixel erscheinende Laserlinie zu berücksichtigen, wird die Basis des Samples zu $2\delta + 1$ vergrößert.

Es wird zunächst der Mittelwert m der Helligkeitsverteilung der zu bearbeitenden Zeile f bestimmt; der zu minimierende Korrelationswert $r(y)$ bestimmt sich dann als

$$2 \cdot (f(0) - m) - (f(\delta) - m) - (f(-\delta) - m)$$

. Als günstiger, für alle praktisch vorkommenden Entfernungen der Laserlinie geeigneter Wert wurde bei einer Auflösung von 320 x 240 Pixeln $\delta = 3$ heuristisch bestimmt, für 640 x 480 Pixel $\delta = 6$.

Subpixel-Genauigkeit

Die Anwendung eines Laser-Peak-Detektors mit Subpixel-Genauigkeit ermöglicht eine Interpolation zwischen LUT-Werten, womit man vor allem bei entfernten Objekten im Bereich niedriger Tiefenauflösung eine deutliche Verbesserung erzielt (Abb. 3.10; vgl. Forest u. a. 2004).

Bei einer Helligkeitsverteilung der Bildzeile $h(x)$ gelte

$h'(x) = h(x) + \frac{h(x+1)}{2} - h(x-1) - \frac{h(x-2)}{2}$. Dann ist der subpixel-genaue Peak \hat{x} zum pixelgenauen Peak x

$$\hat{x} = \begin{cases} x, & \text{für } h'(x) = 0 \\ x - 0.5 + \frac{h'(x)}{h'(x) - h'(x-1)}, & \text{sonst.} \end{cases} \quad (3.3)$$

Damit kann die einem subpixelgenauen Laserpeak \hat{x} in Bildzeile y zuzuordnende Weltkoordinate P bestimmt werden als

$$\begin{aligned}
y &\leftarrow \text{Bildzeile} \\
x' &\leftarrow \hat{X} \mod 1 \\
X' &\leftarrow \hat{X} - x' \\
P &= LUT(X', y) + x' \cdot (LUT(X' + 1, y) - LUT(X', y))
\end{aligned} \tag{3.4}$$

Bei der Betrachtung der Ergebnisse ist zu beachten, dass die Subpixel-Peakdetektion im Bild nur spaltenweise ausgeführt wird. Das heißt, man wird bei der räumlichen Interpolation eine Verbesserung der Ergebnisse in der Dimension oder den Dimensionen erhalten, die mit dieser Bildachse korrespondieren. Dies ist die Höhendimension. Für die seitliche Dimension spielt die Interpolation keine Rolle, da diese durch die diskrete Drehung des Schrittmotors um MA bestimmt wird; aber auch in der Entfernungsdimension wird kein nennenswerter Interpolationseffekt auftreten, da bei ungefähr orthogonal zum Boden stehender Laserebene L die interpolierte Bildachse nur einen Bruchteil an Information zur Tiefendimension beiträgt.

Als Resultat liegen z.B. die nicht-interpolierten 3D-Punkte eines Laserschnittes durch ein zum Boden orthogonales, planes Objekt nicht auf einer Linie. Dies widerspricht nur scheinbar dem Kollinearitätsprinzip, also der zwingenden Abbildung von Geraden auf Geraden. Der Effekt entsteht durch die nichtlineare radiale Kameraverzerrung und die diskrete Natur der LUT (vgl. Abb. 3.9). Beim Erstellen derselben wurde jedem Bildpunkt sein entzerrtes Äquivalent zugeordnet und für diesen der korrekte Weltschnitt berechnet. Dies bedeutet, dass Pixel einer Spalte oder Zeile im Randbereich des Bildes in entzerrter Form nicht auf einer Gerade liegen; folglich liegen auch die korrespondierenden 3D-Punkte eines Laserschnittes nicht auf einer Geraden (Ausnahme sind die Punkte der Bildzeile und Bildspalte durch den Hauptpunkt c). Diese Verzerrung, die sich immer innerhalb der Grenzen eines Schrittes der Tiefenauflösung bewegt, wird, wie im vorigen Absatz erklärt wurde, auch bei Supixel-Peakdetektion und 3D-Interpolation nicht korrigiert.

Für die beabsichtigte Abstraktion der 3D-Punktwolke in eine Draufsicht, wie sie den im nächsten Kapitel dargestellten Navigationsalgorithmen zugrundeliegen, ist diese Interpolation dennoch ausreichend. Wie in Abbildung 3.10 zu sehen ist, werden durch die Interpolation über die Höhendimension die 3D-Punkte so verteilt, dass sich ihre Verteilung in der Draufsicht einer Linie mit Länge der Tiefenauflösung an der entsprechenden Stelle annähert. Für die Zwecke dieser Arbeit ist dies völlig ausreichend.

3.3.6 Hinzufügen von Zufallswerten

Während die mit der Entfernung abnehmende Auflösung des Sensors in der Tiefen- und Höhendimension durch die subpixelgenaue Erkennung des Laserpeaks verbessert wird, bleiben in der lateralen Dimension (bezogen auf die einzelnen Laserschnitte) Lücken zwischen den Sensorpunkten. Dies ist bedingt durch die diskreten Schritte, in denen die Laserebene im Raum gedreht wird.

Die Größe dieser Lücken nimmt linear mit der Entfernung zu und sie wird kritisch für einen auf durch Klassenbildung über die Höhendimension erstellten 2D-Karten operierenden Navigationsalgorithmus, sobald sie eine Größenordnung erreicht, bei der sie sich in der Umgebungskarte widerspiegelt.

Dies ist bei den in dieser Arbeit gewählten Parametern (insbes. Bildauflösung 320 x 240) ab Entfernungen von ca. 3 Metern der Fall. Diesem Effekt wird entgegengewirkt, indem vor der Umrechnung der einzelnen Schnitte nach 3D eine linear von der Entfernung abhängige Zufallskomponente zur seitlichen Position jedes Punktes addiert wird. Damit gilt für jeden Punkt $p = (x, y, z)$: es ist $p' = (x + \delta_x \cdot z, y, z)$, mit $\delta_x = 2Z \cdot \sin(\frac{\pi}{400})$, mit Z als gleichverteilter Zufallsvariable aus dem Intervall $[-1..1]$.

3.4 Ergebnisse

3.4.1 Nahbereich

Im Nahbereich werden aus 10 Scans bei einer Kameraauflösung von 320 x 240 jeweils drei Punkte im Raum bestimmt. Das Messobjekt ist ein Quader mit bekannten

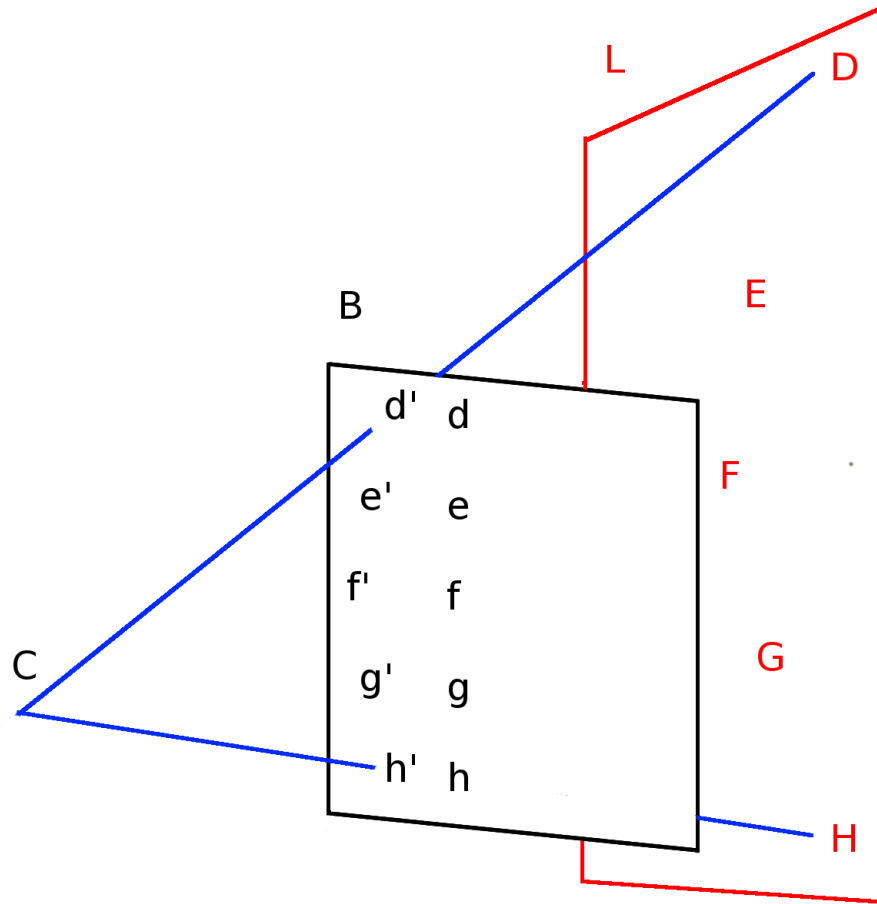


Abbildung 3.9: Nicht-interpolierte 3D-Koordinaten sind wegen der radialen Kamera-verzerrung nicht kollinear. Verzerrungsbehaftete Bildpunkte $d \dots h$ einer Pixelspalte; enzerzte Bildpunkte $d' \dots h'$; resultierende Raumpunkte $D \dots H$ in Laserebene L ; Projektionszentrum C .

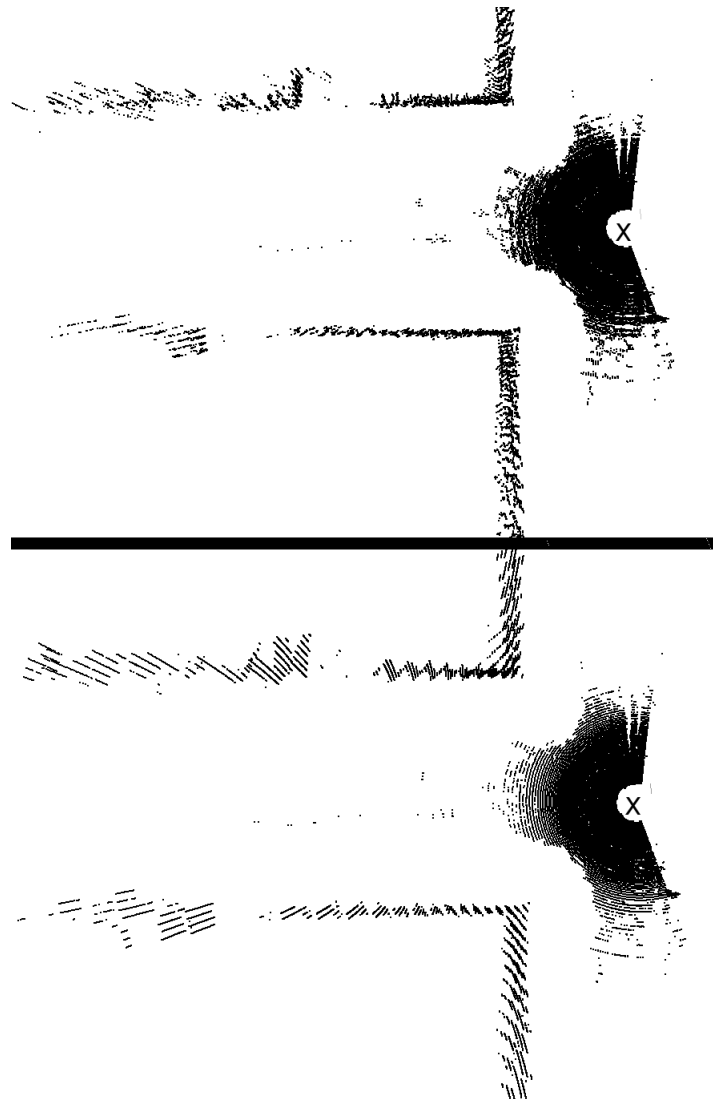


Abbildung 3.10: Draufsicht auf eine Punktwolke aus dem Scan einer Flurgabelung, Kameramodus 320 x 240. Bild oben mit, unten ohne Subpixelinterpolation. X ist jeweils der Standort des Scanner, darum liegend die Standoff Distance als kleiner, runder, leerer Bereich, darum liegend die Punkte des erkannten Bodens; die anderen Punkte sind Wände.

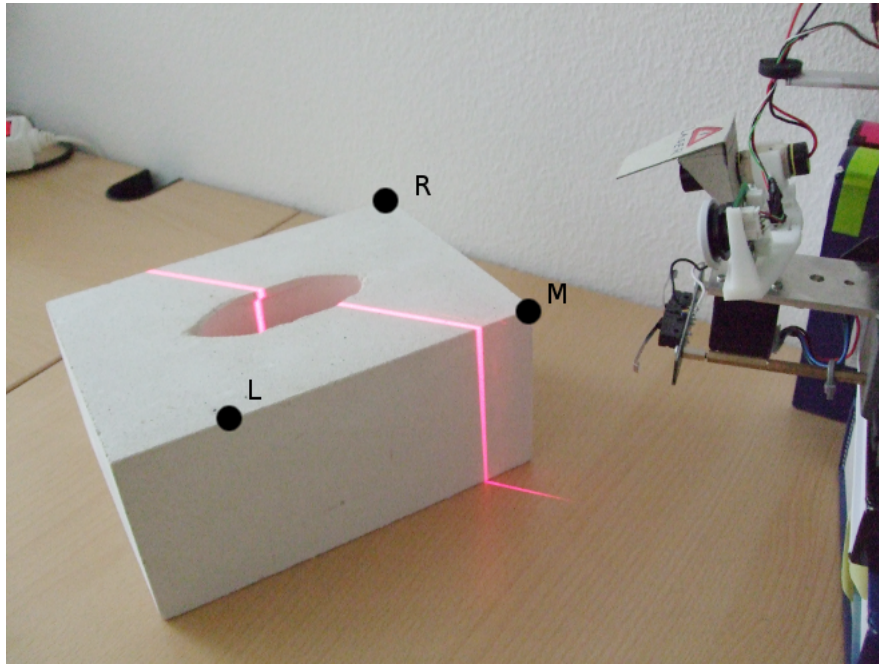


Abbildung 3.11: Anordnung der Punkte L , M , R der Nahbereichsmessung.

Dimensionen. Vermessen werden drei benachbarte Ecken L , M , R , vgl. Abb. 3.11. Von M ist die absolute Lage im Raum bekannt, die nahe an der Mindestreichweite des Sensors platziert wird. Über diesen Punkt soll die Genauigkeit in absoluten Koordinaten bestimmt werden. Aus der Entfernung $|M, R|$ soll die Streckentreue, aus dem Winkel $\kappa(L, M, R)$ die Winkeltreue bestimmt werden. Wegen des nach unten gerichteten Sichtfeld des Sensors wird dieser gegenüber dem Messobjekt erhöht aufgestellt. Die resultierende Punktwolke der Scans wird zur Auswertung im STL-Format gespeichert. Bei der Auswertung der Punktwolken fiel auf, dass alle Scans ein leicht verschobenes Doppelbild des Objekts erzeugten. Dies macht eine automatisierte Auswertung schwierig, darum wurden die Punktwolken in die Software Blender importiert und die gesuchten Punkte manuell ausgewählt. Als Ergebnis ist festzuhalten:

Die Entfernung von M vom Fahrzeugkoordinatensystem wird mit durchschnittlich 19.9 cm bestimmt, wobei extern 21.3 cm gemessen wurden. Dies entspricht einem systematischen Fehler von 7% in der Entfernungsbestimmung (es wird zu kurz

gemessen). Der zweite Punkt des oben genannten Doppelbildes lag etwas näher am Ziel. Die maximale Abweichung in den einzelnen Koordinatenachsen lag jeweils unter einem Zentimeter. Die Strecke $|M, R|$ wurde mit durchschnittlich 15.7 cm um rund 10% zu kurz bestimmt. Der Winkel L,M,R, real 90° , wurde mit durchschnittlich 91.8° bestimmt.

3.4.2 Weitbereich

Zur Messung im Weitbereich wurde die Entfernung zu drei Wänden bestimmt und diese gemessen. Diese Messung erfolgte nur auf der Ebene der Navigationskarten, d.h. mit Kartendarstellungen von einer Schachbretteinheit = 3.50 cm. Es wurde eine Messung bei hoher und eine bei niedriger Kameraauflösung durchgeführt (640 x 480 bzw. 320 x 240). Das Messergebnis hoher Auflösung wurde zusätzlich mit auf halbe Größe skalierten Bildern berechnet.

Im Ergebnis wurde die Entfernung der Wände nur beim genuine Scan mit hoher Auflösung korrekt. Die Entfernungen von 3 m = ca. 86 EH sowie zweimal 1.19 m = ca. 34 EH lassen sich hier korrekt ablesen.

Sowohl beim genuine mit geringer Auflösung aufgenommenen als auch beim auf geringe Auflösung herunterskalierten Ergebnis sind Abweichungen festzustellen.

Beim herunterskalierten Scan ist der Fehler geringer; die gemessenen Entfernungen sind hier mit 32 bzw. 82 EH rund 5 % zu kurz. Beim genuine niedrigauflösenden Scan sind die gemessenen Entfernungen mit 31 bzw. 77 EH rund 10% zu kurz.

3.4.3 Bewertung

Im Weitbereich konnte für den Modus 640 x 480 eine korrekte Abbildung festgestellt werden. Dagegen lieferte der Sensor im zur Navigation verwendeten Auflösungsmodus von 320 x 240 Pixeln systematisch falsche Entfernungswerte. Als Ursache ist die nicht ausreichend kritische Anwendung der bei hoher Kameraauflösung ermittelten Verzerrungsparameter K auf die Abbildung bei niedriger Kameraauflösung zu vermuten.

Als Ursache für die Doppelbilder wurde bestimmt, dass der Peakdetektor für die im Nahbereich besonders dick erscheinende Laserlinie zufällig verteilt zwei Abschnitte zurückgibt, die deren linken und rechten Rand markieren. Da dieser Effekt mit der in weiter entfernten Bereichen kleiner erscheinenden Laserlinie verschwindet, ist die so entstehende Mehrdeutigkeit auf den Nahbereich beschränkt und wird bei zunehmender Entfernung verschwinden, der Fehler wird also nicht mit der Entfernung zunehmen.



Abbildung 3.12: Scan eines Fahrrads in verschiedenen Ansichten; Hintergrund manuell freigestellt.

Problematisch sind Umgebungen stark uneinheitlicher Helligkeit, insbesondere in Verbindung mit direktem Sonnenlicht. Die Kamera verfügt nicht über den notwendigen Belichtungsumfang, um beides gleichzeitig zu erfassen. Ein Betrieb bei auch relativ hellem Kunstlicht ist demgegenüber unproblematisch. Auch spiegelnde Materialien

können zu falsch positiven und falsch negativen Hindernisdetektionen führen; insbesondere ist die Reichweite, mit der ein befahrbarer Untergrund erkannt werden kann, auf spiegelnden Bodenmaterialien deutlich verringert.

Die Verarbeitungsgeschwindigkeit der Bilddaten kann softwaretechnisch noch deutlich erhöht werden, wofür im Rahmen der vorliegenden Arbeit keine Zeit blieb.

Insbesondere sollte mit der Filterung, dem Tabellen-Lookup und der Interpolation aus jedem Bild begonnen werden, sobald dieses im PC vorliegt. Dies wird erreicht, indem man die entsprechenden Funktionen in eigene Prozesse kapselt (fork), anstatt wie in der gegenwärtigen Implementierung auf das vollständige Ende der Aufnahmen zu warten.

Um die Verarbeitungsgeschwindigkeit insbesondere des Peakfilters weiter zu erhöhen, kann man die zu filternden Bilddaten zeilen- oder spaltenweise reduzieren und damit auf die über den Anforderungen der Navigation liegende, sub-Milimeter-genaue Auflösung im extremen Nahbereich verzichten; da jedoch keine Dimension des Bildes allein einer Raumdimension zuzuordnen ist, ist die Entscheidung, welche Zeilen oder Spalten verworfen werden sollen, nicht einfach zu treffen.

Das vor allem bei Kameras mit CCD-Technologie verbreitete Rolling-Shutter-Phänomen (Nicklin u. a., 2007) konnte beim hier verwendeten, auf der CMOS-Technologie basierenden Kameramodul nicht festgestellt werden. Bei einer Aufnahme mit Rolling-Shutter-Verzerrung führen relative Bewegungen von Kamera und aufgenommener Szene zu einer trapezförmigen Bildverzerrung, da ein CCD-Gitter zeilen- oder spaltenweise zeitverzögert ausgelesen wird. Ein Test mit Variationen von Drehsinn und -Geschwindigkeit des Scans ergab hier keine entsprechenden Verzerrungen.

4 Autonomes System

In diesem Kapitel wird zunächst die Hardware- und Softwarearchitektur des entwickelten autonomen Systems beschrieben; es folgt eine Beschreibung des implementierten Weltmodells, der Abstraktionen der Scannerdaten, der Navigationsalgorithmen und der Lokalisierung. Das Kapitel abschließend werden die Resultate mehrerer autonomer Fahrten in einer Büroumgebung vorgestellt und interpretiert.

4.1 Systemarchitektur

4.1.1 Hardware

Neben dem im vorigen Kapitel beschriebenen 3D-Sensor sind seine Hauptkomponenten ein Netbook *ONE A 110* (identisch mit Quanta IL-1) und ein mikrocontrollergesteuertes, mobiles Robotersystem *Conrad/Arexx RP-6*. Daraus wird der Name des hier entwickelten autonomen Systems abgeleitet als “ARP-117”. Die Auswahl des Rechners erfolgte vor allem nach Kostengesichtspunkten. Verglichen mit Ein-Platinen-PCs ist das Gerät schwer und sperrig, jedoch bietet es für seinen geringen Preis von ca. 150 Euro eine vergleichsweise hohe Rechenleistung, umfangreiche Peripherie und eine eigene Stromversorgung. Der RP-6 ist bei einem Preis von ca. 130 Euro eine der günstigsten Roboterplattformen auf dem Markt. Seine Steuerung erfolgt über einen Mikrocontroller ATMEL ATmega. Seine Hardware umfasst zwei frei steuerbare Raupenantriebe mit jeweils eigenem Odometrie-Encoder; Kollisionsschalter; eine vereinfachte Programmierstelle über einen MC-Bootloader und eine UART-Schnittstelle; Stromversorgung und Spannungsmesser. Alle Funktionen können

über eine mitgelieferte C-Programmbibliothek angesprochen werden. In seinem Auslieferungszustand ist der RP-6 darüberhinaus mit einfachen Infrarotsensoren und einer Schnittstelle zur Infrarotkommunikation ausgestattet.

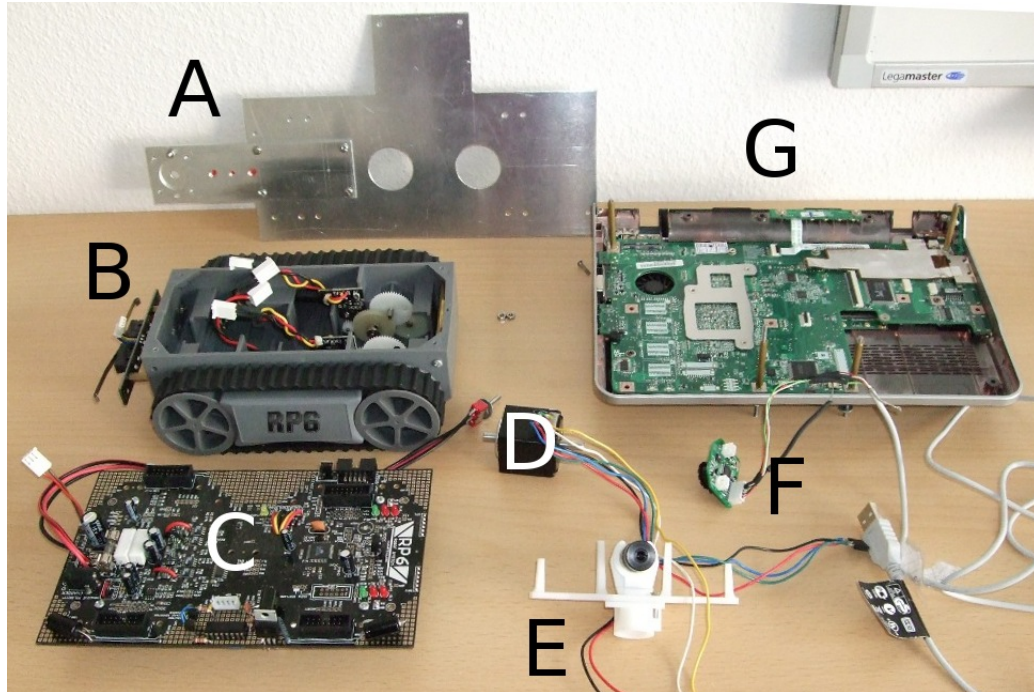


Abbildung 4.1: Hardwarekomponenten des ARP-117. **A** Trägerplatten, **B** Chassis RP-6, **C** Kontrollplatine RP-6, **D** Schrittmotor für Sensor, **E** Laser und Prototyp der Sensoraufnahme, **F** Kamera, **G** Netbook One A110 mit Risern

Die Komponenten werden folgendermaßen modifiziert: Die Kontrollplatine wird vom Chassis des RP-6 entfernt und durch eine Trägerkonstruktion aus Aluminiumblechen ersetzt. Diese trägt auf Risern die Steuerplatine und das Netbook. Von letzterem werden Tastatur, Touchpad, Bildschirm und oberer Gehäusedeckel entfernt; die Unterseite liegt oben, so dass die Elektronik geschützt ist. Sämtliche Aufbauten wurden so platziert, dass eine ausgeglichene Gewichtsverteilung besteht. Die elektrischen Verbindungen der RP6-Platine werden mit Steckverbindern ausgestattet. Der Platine hinzugefügt wird eine Treiberstufe zur Ansteuerung des Schrittmotors und ein MOSFET zum Schalten des Lasers. Die Infrarot-Entfernungsmesser des RP6 werden

entfernt. Die Treiberstufe des Schrittmotors wird über digitale Ausgänge des Atmel-MC angesprochen. Die beiden Eingänge, auf denen beim RP-6 die Kollisionsschalter (“Bumper”) liegen, werden zusammengefasst; der frei gewordene Eingang wird zum Anschluss des Schrittmotor-Endschalters genutzt. Schrittmotor und Sensorverbund werden vor dem Fahrzeug angeordnet; die Bumper werden mit Risern vor den Lichtschnittsensor verlegt. Um die UART-Verbindung, die zur Programmierung des MC verwendet wird, frei verwenden zu können, muss ein Schalter in die Reset-Leitung RTS/CTS eingelötet werden, da diese im Auslieferungszustand zum Reset des MC in Zusammenhang mit der Bootloaderprogrammierung verwendet wird. Als Konsequenz kann im Folgenden kein UART-Hardware-Handshake verwendet werden, sondern die Kommunikation muss mit explizitem Senden von XON / XOFF synchronisiert werden.

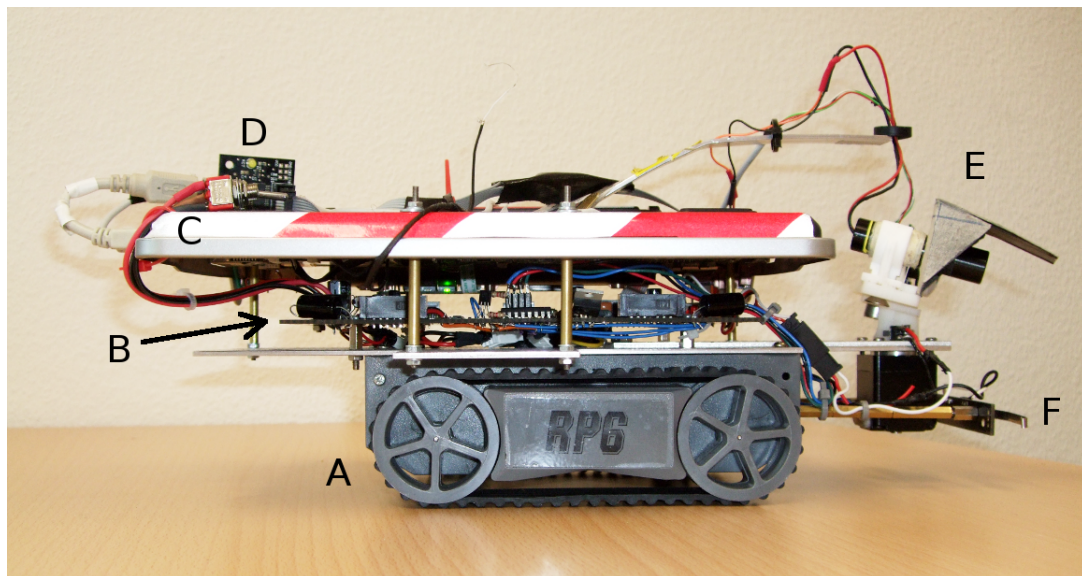


Abbildung 4.2: Das fertig aufgebaute autonome System ARP-117. (A) Chassis, (B) Steuerungsplatine, (C) Netbook, (D) USB/UART-Wandler, (E) Lichtschnittsensor, (F) Kollisionsschalter.

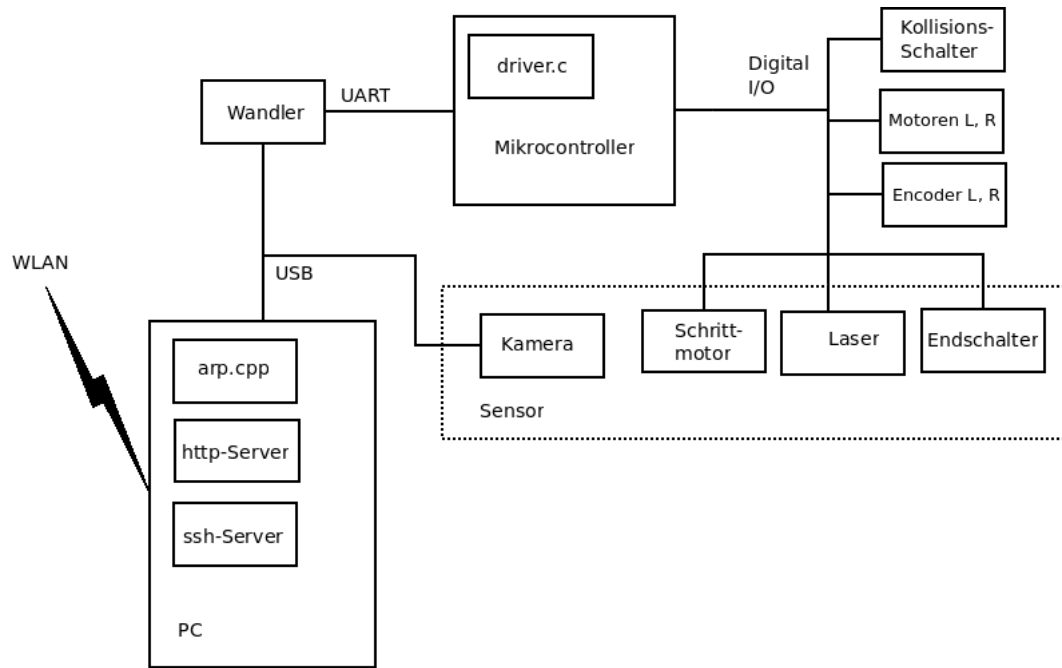


Abbildung 4.3: Softwarearchitektur

4.1.2 Software

Die implementierte Softwarearchitektur basiert auf einer Arbeitsteilung zwischen Mikrocontroller und PC. Dabei übernimmt der MC die unmittelbare Hardwareansteuerung, also: die Regelung der Fahrmotoren, das Auslesen der Odometrie-Encoder, das Registrieren von Kollisions- und Endschalterkontakten, das Schalten des Lasers und die Ansteuerung des Schrittmotors. Auch wenn in diese Arbeit keine formellen Echtzeitbetrachtungen eingegangen sind, kann festgestellt werden, dass es nur drei echtzeitrelevante Funktionen zur Abwehr von Schäden an System oder Umwelt gibt: Stopp der Fahrmotoren bei Auslösen der Kollisionsschalter; Stopp des Schrittmotors bei Auslösen des Endschalters; und Abschalten des Systems bei einer Störung der Stromversorgung. Alle diese Funktionen sind MC-seitig Interrupt-gesteuert; dies gilt ebenfalls für die Regelung der Fahrmotoren.

Auf diese Weise wird der PC von Echtzeitanforderungen freigehalten und kann rechenintensive oder blockierende höhere Funktionen ausführen: Bildverarbeitung, Wegplanung, Sensorintegration, Kommunikation mit einer Kontrollstation. Da keine Linux-Treiber zur Verfügung stehen, um den WLAN-Chip des A110 im Ad-Hoc-Modus zu betreiben, muss der kontrollierende PC entweder als Access Point fungieren, oder die Kommunikation über einen WLAN-Router abgewickelt werden.

4.1.3 Odometrie

Der RP6 verfügt über zwei Encoder, die die Umdrehungen der Antriebe mit einer Auflösung von ca. $\frac{1}{4}$ mm auflösen. Dennoch ist die Genauigkeit der Odometrie für eine korrekte Integration einer Folge von mehreren, bei autonomer Fahrt aus verschiedenen Positionen aufgenommenen Scans nicht ausreichend. Die Ursache liegt an den unvorhersehbaren Reibungskräften, die auf den Raupenantrieb wirken. Große Genauigkeitsschwankungen ergeben sich besonders bei Drehungen auf der Stelle und dem resultierenden seitlichen Schieben über die Gummiraupen.

Man kann durch eine Reihe von Messungen die relative Rotationsgeschwindigkeit und -entfernung des linken und rechten Antriebs so justieren, dass die Drehung im Mittel dem gewünschten Winkel entspricht; dennoch sind auch auf günstigem Untergrund Abweichungen von im Einzelfall bis zu $\pm 20\%$ unvermeidlich. Auf besonders glatten oder besonders rauen Untergründen kann die Rotation auch noch schlechter ausfallen bzw. völlig versagen.

Demgegenüber zu vernachlässigen ist der Fehler in der Geradeausfahrt. Bei dieser beschreibt der Roboter einen großen Kreis; experimentell festgestellt wurde ein Radius von 55 m, auch wenn sich dieses Ergebnis nicht verallgemeinern lässt. Dieser Fehler lässt sich nicht durch Programmierung im Mikrocontroller korrigieren, da die Auflösung der Eingabewerte des die Motoren steuernden A/D-Wandlers zu grob ist.

Ein Lösungsansatz zur Verbesserung der Drehung wäre es, in Kreisbögen zu fahren; so würde durch das Vermeiden von Drehungen auf der Stelle die Reibung quer zu den Antriebsraupen reduziert und das Ergebnis vermutlich voraussagbarer. Hier wären

jedoch komplexe Algorithmen für die Wegplanung erforderlich; solche sollen in dieser Arbeit nicht behandelt werden, denn auch mit einer solchen Steuerung wäre der Positionsfehler letztendlich nicht zuverlässig vorausszusagen.

Darum wurde beschlossen diesen Fehler zu belassen und gegebenenfalls zu modellieren. Das Problem zusammenfassend sind also Methoden erforderlich, um die Positionsbestimmung über die Auswertung der Odometrie hinaus auch durch Auswertung der Sensordaten zu verbessern. Eine solche Methode wird im Abschnitt 4.2.6 entwickelt.

4.2 Weltmodell

Die in einem Scandurchlauf vom Sensor erzeugte Punktwolke umfasst bei der hier verwendeten Kameraauflösung von 320×240 und 230 Laserschnitten bis zu $320 \cdot 230 = 73600$ 3D-Punkte; auf einer solchen Datenmenge lassen sich Navigationsalgorithmen nur schwer ausführen.

In dieser Arbeit wird statt dessen die 3D-Punktwolke auf einfachere Repräsentationen reduziert. Es werden zunächst zwei 2D-Karten direkt aus der Punktwolke abgeleitet: eine Karte M_w als klassisches “Occupancy Grid”, also als Karte von Hindernissen; und eine Karte M_f des als befahrbar erkannten Bodens.

Nullpunkt jeder Punktwolke ist O_F , der Schnittpunkt der Sensor-Rotationsachse mit einer idealen Bodenebene (vgl. 3.4). Dies ist auch der Nullpunkt aller von einem Scan erstellten zweidimensionalen Repräsentationen; ihr Koordinatensystem ist egozentrisch. Die Gesamtheit der Repräsentationen eines einzelnen Scans wird im folgenden als ein *Frame* bezeichnet. Die einzelnen Frames werden bei der Kombination zur Gesamtkarte in ein allozentrisches Koordinatensystem überführt; dessen Nullpunkt 0_M entspricht 0_F im ersten erstellten Frame. Alle folgenden Frames werden auf den neuen Standort des Roboters verschoben, rotiert und der Gesamtkarte *Map* hinzugefügt, die wiederum die schon in den Frames vorhandenen, verschiedenen Ebenen von Repräsentationen umfasst.

Die Auflösung der Karten wird auf eine Sensoreinheit festgelegt, also die Größe der Schachbrettmuster, die zur Sensorkalibrierung verwendet wurden. In dieser Implementierung sind das 3.50 cm pro Einheit.

Welcher Vorteil des 3D-Sensor bleibt bei dieser Informationsreduktion? Erhalten bleibt die Erkennbarkeit von Hindernissen, die mit einem 2D-Entfernungsmesser nicht detektiert würden, wenn sie nicht in seiner Scanebene liegen, also Senken genauso wie Objekte, die unterfahren werden könnten.

Um die Punktwolke in die genannten Gitter M_w und M_f zu überführen, wird die vereinfachende Annahme getroffen, dass sich das Fahrzeug in einer dreidimensionalen Welt, dabei aber stets auf einem ebenen Untergrund bewegt. Auf diese Weise kann eine Zuordnung von 3D-Scan-Punkten zu 2D-Punkten der Boden- und der Wandkarte durch einfache Klassenbildung in der Hochachse und eine anschließende Schwellwertoperation vorgenommen werden (minimale Anzahl von 3D-Punkten pro 2D-Gitterpunkt). Eine Ausnahme von der Klassenbildung ist, dass 3D-Punkte auf der Höhe von Hindernissen dazu führen, dass der Gitterpunkt des befahrbaren Bodens M_f für diesen Durchgang gelöscht wird. So werden Positionen, die sowohl 3D-Punkte in der Höhe von Hindernissen als auch in der Höhe des Bodens aufweisen, gelöscht; dies betrifft Überhänge und allgemein Hindernisse die sich auch auf die Höhe des Bodens erstrecken. Da das Gesichtsfeld des Sensors nach oben hin stark begrenzt ist werden durch letztgenannte Maßnahme Punkte des befahrbaren Bodens nur dann ausgeschlossen, wenn sie nahe der Bauhöhe des Fahrzeugs liegen, und nicht etwa durch beliebig hohe Punkte z.B. an der Raumdecke.

4.2.1 Bereiche von Interesse

Von den Grundabstraktionen M_w und M_f werden nun weitere Ableitungen erstellt. Zunächst eine Karte des "offenen Raums" M_{space} , die sich ergibt, indem für jedes gesetzte Pixel in der binären Hinderniskarte M_w eine 8-konnektierte Linie (also ein approximiertes Rechteck) in der neuen Binärkarte M_{space} zum Zentrum des Scans gezogen wird. Aus M_{space} werden potentielle Störungen durch einzelne, fehlerhaft

Algorithmus 3 Grundabstraktionen: Hinderniskarte (Wandkarte) M_w , Bodenkarte (“floor”) M_f

```

for  $\forall p_w \in M_w, \forall p_q \in M_f$  do
     $p_w \leftarrow 0, p_q \leftarrow 0$ 
end for

for  $\forall p \in \text{Punktwolke}$  do
    if  $M_f(p.x, p.y) \neq -1$  then
        if  $p.z > \min_{Boden}$  UND  $p.z < \max_{Boden}$  then
             $M_f(p.x, p.y)++$ 
        end if
    end if

    if  $p.z > \min_{Wand}$  then
         $M_w(p.x, p.y)++$ 
         $M_f(p.x, p.y) = -1$ 
    end if
end for

for  $\forall p_w \in M_w, \forall p_q \in M_f$  do
    IF  $p_w > \text{threshold}_w$  THEN  $p_w \leftarrow 1$  ELSE  $p_w \leftarrow 0$ 
    IF  $p_f > \text{threshold}_f$  THEN  $p_f \leftarrow 1$  ELSE  $p_f \leftarrow 0$ 
end for

```

erkannte Wandpixel reduziert, indem aufeinanderfolgend zwei morphologische Operationen ausgeführt werden: zunächst eine Erosion, danach eine Dilatation mit je $\delta_x = \delta_y = 3$ (heuristisch bestimmt).

Im nächsten Schritt werden die Kanten von M_{space} mit einem Canny-Filter bestimmt. Die resultierende Karte ist M_{border} . Zieht man von M_{border} nun die einer Dilatationsoperation unterzogene Karte $M'_{wall} = dilate(M_w)$ ab, erhält man eine neue Karte $M_{interest}$. Diese enthält jene Grenzen des vom gegenwärtigen Standort einsehbaren Bereichs, die sich nicht unmittelbar durch Hindernisse ergeben, sondern durch vor ihnen liegende Verdeckungen. Dies sind genau jene Grenzen des bekannten Raumes, an denen ein Potential für weitere Erkundung besteht.

4.2.2 Indikatoren

Als nächstes ist eine Karte gesucht, die den Raum danach klassifiziert, wo sich gute Standorte zum Einsehen der durch $M_{interest}$ markierten Verdeckungen befinden. Dazu wird $M_{interest}$ mit Hilfe einer probabilistischen Hough-Transformation abstrahiert. Die so gewonnenen Liniensegmente (p, q) werden in flächige Kartenobjekte, genauer: in Quadrate $QI = (p, a, q, b)$ mit den Liniensegmenten als Diagonale transformiert, die in einer Karte als Zielindikatoren dienen (s. Abb. 4.4). Es gilt:

$$\begin{aligned} a &= \begin{pmatrix} cx + hy \\ cy - hx \end{pmatrix}, b = \begin{pmatrix} cx - hy \\ cy + hx \end{pmatrix} \text{ mit} \\ h &= \begin{pmatrix} hx \\ hy \end{pmatrix} = \frac{1}{2}(q - p) \text{ und } c = \begin{pmatrix} cx \\ cy \end{pmatrix} = p + h. \end{aligned} \tag{4.1}$$

Der “Farbwert” eines Polygons wird linear mit der Länge seiner Größe skaliert; unbekanntere Bereiche größerer Ausdehnung erhalten damit bei der Zielsuche ein höheres Gewicht.

Die Navigationskarte wird insgesamt mit einem Wert initialisiert, der einem Polygon der Breite $b = 1$ entspricht. So kann man die Navigationskarte mit einem Binärbild der befahrbaren Bereiche minimumverknüpfen und behält so eine Unterscheidung zwischen

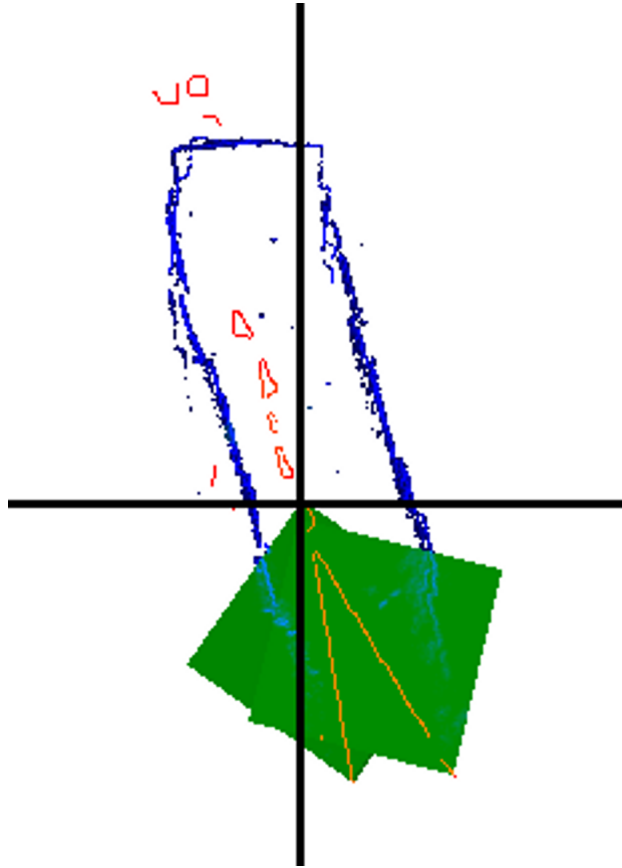


Abbildung 4.4: Aus mehreren Scans und Odometriedaten zusammengesetzte Karte. Standort des Fahrzeugs schwarz markiert. Blau: Flurwände mit Vorsprüngen und Türrahmen, rot $M_{interest}$ (Grenzen des einsehbaren Bereichs, die nicht durch erkannte Objekte hervorgerufen werden; inkl. Residuale aus vorherigen Scans), grün: Indikatoren. Das autonome System wird in die Richtung navigieren, in der ein Schnitt durch die Indikatoren maximiert wird.

nicht befahrbaren Bereichen $f(\min(\text{Navigationskarte}, \text{Bereich}) = 0)$, befahrbaren Bereichen ohne ($f = 1$) und mit besonderem Interesse ($f > 1$).

Fügt man jeden dieser Bereiche einer Karte $M_{\text{interestArea}}$ hinzu, steht deren Maximum für jene Position im Raum, von der aus die meisten unbekannten Bereiche eingesehen werden können.

Das Hinzufügen selbst ist formal ein Maximum-Operator; zeitgünstig ist es, die QI nach aufsteigender Größe zu sortieren und sie in dieser Reihenfolge überschreibend der Karte hinzuzufügen.

4.2.3 Navigationskarten

Bevor nun mit $M_{\text{interestArea}}$ navigiert werden kann, muss der als mit Sicherheit befahrbare Boden aus der Karte M_{floor} in Betracht gezogen werden. Wie im Sensor-Kapitel dargestellt, kann der Boden nur auf geringere Entfernung erkannt werden als über den Boden erhabene Hindernisse (s. Abb. 3.8). Dabei stößt man bereits, abhängig vor allem von den optischen Eigenschaften des Bodenmaterials, bereits in noch geringen Entfernungen auf Bereiche, in denen der Boden zwar erkannt wird, dies aber nur lückenhaft. Um eine sichere Navigation zu gewährleisten, werden aber nur Pfade durch lückenlose Bereiche in M_{floor} berücksichtigt, das Verfahren erzeugt somit nur einen relativ kleinen zusammenhängenden Bereich.

Um diesem Effekt entgegenzuwirken, wird die Auflösung des Bodengitters um den Faktor zwei verringert. Dabei wird die Auflösung der Bodenkarte beibehalten, um einen einfacheren Austausch mit den anderen Karten zu ermöglichen; jedoch wird das Bodengrid "relaxt" nach Algorithmus 4:

Algorithmus 4 "relax" M_{floor}

```

for  $p_{x,y} \in M_f$  do
  if  $p_{x,y} = 1$  then
     $p'_{x,y} \leftarrow p'_{x+1,y} \leftarrow p'_{x,y+1} \leftarrow p'_{x+1,y+1} \leftarrow 1$ 
  end if
end for

```

Wegen dem nicht identischen Rotationszentrum von Sensor und Fahrzeug kann dieses nicht bis unmittelbar bis an ein Hindernis heranfahren kann, insbesondere wegen dem abweichenden Drehzentrum von Roboter und Sensor. Dazu wird auf der "relaxten" Karte wiederum eine morphologische Erosion durchgeführt mit einem Radius, der mit dem maximalen Radius des Fahrzeugs um seinen Drehpunkt korrespondiert. Dies liefert die Karte des sicher navigierbaren Bereichs, $M_{reducedFloor}$.

Es werden zwei Zielkarten erstellt, indem man einerseits den bekannten Raum, andererseits den bekannten Boden mit der Interest Area schneidet. Dabei handelt es sich sämtlich um Binärbilder:

$$M_{shortrangetarget} \leftarrow \min(M_{reducedFloor}, M_{interestArea});$$

$$M_{longrangetarget} \leftarrow \min(M_{space}, M_{interestArea})$$

$M_{shortRangeTarget}$ dient im Folgenden zur sicheren Planung des nächsten Bewegungsschrittes; $M_{longRangeTarget}$ zur Bestimmung übergeordneter Navigationsziele.

4.2.4 Wegsuche

Ziel der Wegplanung ist eine effiziente Erfassung unbekannter Areale. Für die Navigation des autonomen Systems gilt allgemein folgendes: Während die Auswahl globaler Ziele auf der gesamten bisher erstellten Karte (Map) erfolgt, wird der tatsächlich befahrene Bereich durch den jeweils letzten Scan (Frame) beschränkt. Dabei erzeugt jeder Wegplanungsschritt genau eine Bewegung, eine Geradeausfahrt gefolgt von einer Drehung. Eine übergeordnete, über mehrere Schritte vorausberechnete Navigation findet nicht statt; allerdings werden durch die Auswahl globaler Ziele aus der gesamten bisher erstellten Karte globale Informationen in die Berechnung jedes Schrittes einbezogen.

Zur Navigation werden zwei Hilfsfunktionen verwendet, die auf $M_{longrangetarget}$ und $M_{shortrangetarget}$ operieren können. Sie bestimmen einen maximalen, geradelinigen Pfad auf diesen. Dazu wird zunächst mit Hilfe des Bresenham-Algorithmus (Linieninterpolation in diskreten Repräsentationen) ein acht-konnektiver Pfad vom Standort p_0 zum Zielpunkt p_1 bestimmt und die Werte aus dem jeweiligen Grid, also

die lokalen Werte der Indikatoren (4.2.2) der im Pfad liegenden Punkte in einem Feld F gespeichert. So lässt sich leicht der Indikatoren-bezogene "Wert" w eines Pfades ermitteln. In der ersten Variante bestimmt die Hilfsfunktion den freien Fahrtrichtungswinkel zur Maximierung von w bis zum Rand der jeweiligen Karte; in der anderen Variante wird beim vorgegebenen Weg eine optimale Strecke gesucht, um w lokal zu optimieren, wobei an der ersten Stelle n mit Wert 0 abgebrochen wird, da ja die Interest-Karten mit 1 initialisiert wurden und ein Wert 0 darum aus der Verknüpfung mit nicht befahrbaren Bereichen resultiert. Die tatsächliche Länge eines solchen Pfades ergibt sich dann als $|p_0 - p_1| \cdot \frac{|F|}{n}$.

Beim Positionsupdate ist zu beachten, dass das Drehzentrum des Fahrzeugs nicht mit dem Zentrum des Sensorkoordinatensystems übereinstimmt. Abgeschätzt aus der Konstruktion des Fahrzeugs liegt das Sensorzentrum 13.84 cm in Fahrtrichtung vor dem Fahrzeugdrehzentrum.

4.2.5 Umgang mit Odometriefehlern

Aus den in Abschnitt 4.1.3 dargestellten Eigenschaften der Odometrie sowie der Entscheidung für ein Stop-and-go-Verfahren ergibt sich der grundlegende Ablauf der implementierten Fortbewegung für den ARP-117. Es wird in Sequenzen Scan - Geradeausfahrt - Drehung vorgegangen. Der Vorteil liegt darin, dass der erheblich größere Fehler durch die Drehung erst nach der Strecken-Ortsveränderung eingeführt wird, wodurch die angenommene Position tendenziell besser mit der tatsächlichen Position übereinstimmt.

4.2.6 Lokalisierung

Kartenupdate

Vorgehalten werden als *Map* drei globale Karten, Wände, Boden, und Space; diese sind Binärkarten mit einer Auflösung von einer Kalibrierungseinheit (Schachbrettfelder s , 3.50 cm). Sie sind so eingerichtet, dass die initiale Pose des Roboters ($x = y = \delta = 0$) im Zentrum der Karte platziert und nach oben ausgerichtet ist. Nach jedem Scan

werden diesen Karten die Repräsentationen des neuen Scans mittels Maximumfunktion hinzugefügt, nachdem diese entsprechend der neuen Pose, durch Odometriemessungen und Scanmatching bestimmt, verschoben und rotiert wurden.

Drehkorrektur

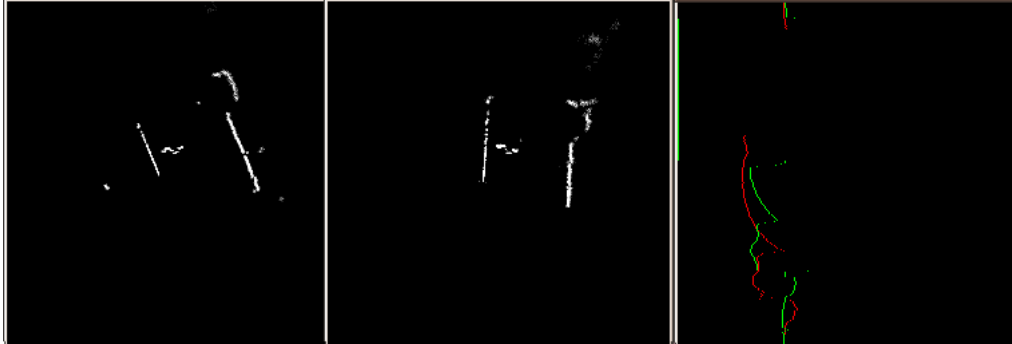


Abbildung 4.5: Mitte: Kartenausschnitt A1, in den nur ein Scan integriert wurde (n_0).
 Links: neuer Scan n_1 nach Vorwärtsbewegung und Drehung, sichtbar ist die höhere Genauigkeit in der rechten "oberen" Ecke des Zimmers. Rechts: log-polar-transformierte Ausschnitte

Um ein Scanmatching durchzuführen, d.h. aufeinanderfolgende Scans nach einer Positionsänderung so aneinander anzupassen, dass eine gegenüber der Odometrie genauere Aussage über die ist-Position getroffen werden kann, sollte bei dem erprobten Verfahren nur die Korrektur der Drehung gematcht werden. Es war vorgesehen, den kartesischen x-y-Bestandteil der Pose weiterhin aus der Odometrie zu gewinnen. Angesichts des vergleichsweise geringen Fehlers bei der Geradeausfahrt schien diese Vorgehensweise plausibel.

Gematcht werden sollte also ein Frame F des letzten Scans mit der Karte K aller bisher erstellten Scans, bei bekannter Translation. Dazu wird ein Ausschnitt gemäß der aktuell angenommenen Position in eine Log-Polar-Darstellung transformiert. Bei der Log-Polar-Darstellung wird jeder kartesischen Koordinate x, y im Urbild eine Koordinate ϕ, d zugeordnet und der entsprechende Farbwert des Urbildes übertragen. ϕ

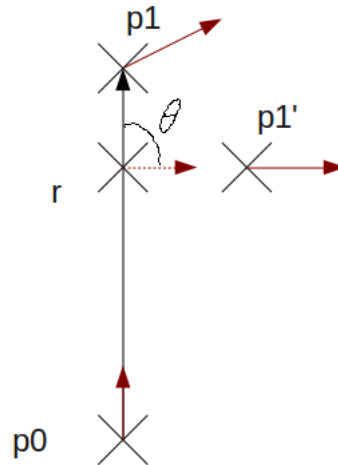


Abbildung 4.6: Navigationsschritt. Jede Pose p_i hat die Elemente x, y , und r . Startpose p_0 , Zielstellung des Navigationsalgorithmus ist p_1 . Geradeausfahrt um $|p_0, p_1|$. Rotationszentrum r bestimmt aus p_1 . Bestimmung der tatsächlichen Rotation θ um r durch Scanmatching (gestrichelter Vektor). Tatsächliche, neue Position p_1' aus p_1 und θ . Der Abstand zwischen Ziel- und tatsächlicher Position ist damit maximal der doppelte Abstand von Sensor- und Roboter-Rotations-Zentrum.

ist dabei der absolute Winkel d der Abstand von (x, y) zum Transformationszentrum lp , wobei d logarithmisch skaliert wird. Die exponentiell abnehmende Tiefengenauigkeit der Log-Polar-Transformation schien dabei gut mit der abnehmenden Tiefenauflösung des Sensors zu korrespondieren.

Um Verdeckungen in der Karte zu berücksichtigen, sind nur Punkte gesucht, die nicht von anderen Punkten verdeckt werden; man nimmt also aus jeder mit einem Winkel ϕ korrespondierenden Zeile die Repräsentation mit minimalem d und bestimmt ihren lokalen Mittelpunkt gemäß dem Algorithmus

```

im log-polar-transformierten Bild:
für jede mit einem Winkel korrespondierende Bildzeile x
    Gehe in Richtung y (zunehmende Entfernung d) bis (x,y) != 0
        startwert <- y
    Gehe weiter in Richtung y bis (x,y) == 0
        stopwert <- y
    Bestimme das gewichtete Mittel der Positionen startwert bis stopwert,
        wobei nicht nach dem Wert, sondern der Position gesucht ist.

```

Die so entstehenden Kurven von Frame und Karte werden durch Verschiebung über die Winkeldimension korreliert (Abb. 4.5).

Beim Scanmatching nach diesem Verfahren sind Mehrdeutigkeiten möglich.

Beispielsweise lässt sich bei einer Sensorposition in der Mitte eines quadratischen Raumes nicht feststellen, ob eine angeforderte Drehung stattgefunden hat oder nicht.

Durch Verkettung (Backtracking) mehrerer Matching-Schritte lassen sich zwar Mehrdeutigkeiten in den weiter zurückliegenden, nicht aber im aktuellen Schritt auflösen. Denkbar wäre als weiterführender Ansatz eine Analyse des Optical flow der Kamera während der Drehung, um einen ungefähren Anhaltspunkt für die tatsächlich erfolgte Drehung zu gewinnen. Als ad-hoc-Lösung wird in dieser Arbeit der Suchbereich auf eine Abweichung von ± 50 der angeforderten Drehung eingeschränkt.

4.3 Ergebnisse

4.3.1 Odometriebasierte Navigation

Erste autonome Fahrt

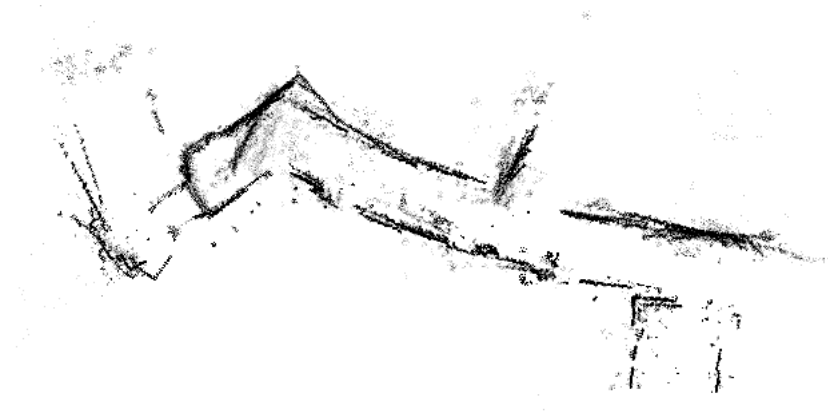


Abbildung 4.7: **Hinweis:** die Abbildungsqualität des Sensors entspricht hier nicht dem Endstand!

Die bei der ersten autonomen Fahrt erstellte Umgebungskarte ist in Abbildung 4.7 zu sehen. Hier ist darauf hinzuweisen, dass der Scanner zu diesem Zeitpunkt noch deutlich schlechter kalibriert war als zum Endstand der vorliegenden Arbeit. Es wurden 28 Einzelscans aus ebensovielen Bewegungsschritten per Odometriedaten kombiniert. Startpunkt war das Büro unten rechts. Der Abbruch des Experiments erfolgte bei einer drohenden Havarie: auf dem besonders ungünstigen Untergrund einer Fußmatte wurde eine angeforderte Drehung weg von einem korrekt als nicht befahrbar erkannten Treppenabsatz nur zu einem Bruchteil ausgeführt. Ohne Scanmatching hatte das System keine Möglichkeit dies zu erkennen und wäre havariert.

Zweite autonome Fahrt

Abb. 4.4 zeigt eine Draufsicht aus der zweiten hier vorgestellten autonomen Fahrt. Gegenüber der ersten Fahrt konnte die Genauigkeit der einzelnen Scans und die

Genauigkeit der Odometriemessungen verbessert werden. Die Fahrt wurde zu einem späteren als dem abgebildeten Zeitpunkt abgebrochen, nachdem das System bei Navigation durch einen Flur keine freien Bereiche zur Weiterfahrt mehr erkennen konnte. Dies kam dadurch zustande, dass durch die sich aufaddierenden Fehler der Odometriemessung das Ende des Korridors in der Karte mit falsch positiven Hindernispunkten belegt wurde, die die Erkennung eines zusammenhängenden freien Bereichs und damit die Konstruktion eines geeigneten Indikators verhinderte. Als Ergebnis ist festzuhalten, dass ein funktionierendes Scanmatching auch dieses Problem beheben würde. Das vollständige Log beider Fahrten befindet sich auf dem der Arbeit beigelegten Datenträger.

4.3.2 Korrigierte Navigation

Das Scanmatching erbrachte in der Implementierung des vorgeschlagenen Verfahrens keine Ergebnisse, die die Zuverlässigkeit der Positionsbestimmung gegenüber einer odometriebasierten Navigation verbessert hätten. Bei der Wahl der Methode wäre es wohl angebracht gewesen, den Hinweis von Bradsky und Kaehler (2004, S. 175) zu berücksichtigen, dass *"such transforms are quite sensitive to the exact location of their center points"*.

Versuche, ein Matching über die gesamte Pose, also sowohl kartesische Koordinate als auch Rotation, mit Hilfe eines Downhill-Simplex-Algorithmus (Nelder und Mead, 1965) zu implementieren, waren in der verbleibenden Zeit nicht mehr erfolgreich. Eine Konvergenz des Nelder-Mead-Algorithmus konnte nur für sehr günstige Startwerte erreicht werden; innerhalb der hohen Schwankungen der Drehgenauigkeit gelang dies nicht. Dennoch scheint ein solches Verfahren prinzipiell geeigneter als das zuerst implementierte Log-Polar-Matching.

5 Fazit

Es wurde aus freier Software und Consumerhardware ein autonomes Fahrzeug mit einem Lichtschnittsensor konstruiert und programmiert, das prinzipiell dazu geeignet ist, selbstständig Innenräume zu erkunden und zu kartieren. Dabei werden auch Senken und Unterfahrungen zuverlässig erkannt.

Dem Prinzip des Laser-Sensors geschuldet können spiegelnde Flächen nur schlecht erkannt werden; auch zwingt die langsame Bildübertragung zum Betrieb der Kamera mit reduzierter Auflösung, was sich ungünstig auf die Tiefenauflösung des Sensors auswirkt. Die Genauigkeit des Sensors ist beim Einsatz in hellem Umgebungslicht, insbesondere Sonnenlicht, beeinträchtigt; eine partielle Verbesserung in der Laserdetektion könnte sich durch die Integration eines auf die Laserfrequenz abgestimmten optischen Filters ergeben; eine grundlegende Verbesserung erfordert eine aufwendigere Kamera mit einer echten Blende.

6 Ausblick

Bei der Navigation autonomer Systeme stellen probabilistische Weltmodelle wie bspw. *Grid-based FastSLAM* nach Thrun u. a. (2006, S. 474 ff) den aktuellen Forschungsstand dar; nur mit diesen ist es effektiv möglich, eine Folge von fehlerbehafteten Positionsmessungen zuverlässig zu integrieren, insbesondere wenn einzelne Messungen für sich allein genommen keine ausreichende Bestimmung des Standorts ermöglichen. Außerdem erlauben sie die Modellierung dynamischer Umgebungen, die in der vorliegenden Arbeit nicht berücksichtigt wurden. Obwohl ein solches Verfahren in diese Arbeit keinen Eingang gefunden hat, scheinen das hier entwickelte System mit seinem 3D-Sensor, seinen Ansätzen zur Modellierung unbekannter Areale, seiner Mobilität und seiner guten Rechenleistung für die Implementierung eines solchen Verfahrens geeignet. Ferner wäre eine Erweiterung der Sensorik durch einen Neigungsmesser vorstellbar, um auch autonome Fahrten in unbefestigtem Gelände in Angriff nehmen zu können.

Abschließend soll eine kurze Betrachtung zur gesellschaftlichen Relevanz der Technologie “autonomer mobiler Roboter” stehen. Am eingangs genannten Beispiel des Staubsaugeroboters zeigt sich, dass deren Einsatzmöglichkeiten - wie die aller Technologien - nicht ohne ihre gesellschaftlichen Rahmenbedingungen betrachtet werden können. Die leistungsfähigsten (und überhaupt effektiv nutzbaren) Staubsaugeroboter für den Heimbereich kosten zur Zeit etwa 2000 Euro, wobei deren kognitive Fähigkeiten immer noch als bescheiden bezeichnet werden müssen. Bei diesem Preis befinden sie sich in einer ungünstigen Konkurrenzsituation gegenüber menschlicher Arbeitskraft, denn Tätigkeiten, für die der Einsatz von Robotern zuerst in

Frage kommt, sind vergleichsweise schlecht entlohnt. Dies gilt auch in Ländern, die im globalen Vergleich ein gutes Lohnniveau aufweisen. Es könnte sogar argumentiert werden, dass eine Verbreitung von autonomen Systemen negative Einflüsse auf den Arbeitsmarkt und damit auf die Lohn- und Lebenssituation großer Gruppen von Beschäftigten haben könnte. Dennoch, wenn man es vom Standpunkt des Gebrauchs- und nicht des Marktwertes betrachtet, erscheint es objektiv wünschenswert, Menschen von repetitiven, schmutzigen oder gefährlichen Tätigkeiten so weit wie möglich zu befreien. Eine Technologie, die geeignet erscheint, den menschlichen Lebensstandard zu erhöhen, soll darum nach Möglichkeit auch dann weiterentwickelt werden, wenn ihr Einsatz in der gegenwärtigen Gesellschaftsordnung an Grenzen stößt.

Anhang

A Laserfokussierung

	0.25	0.5	1	2	3	4
0.25	0.1	0.5	0.7	0.7	0.8	0.9
0.5	0.4	0.1	0.2	0.3	0.4	0.5
1	0.5	0.2	<0.1	0.1	0.2	0.2
2	>0.5	0.3	0.1	< 0.1	0.1	0.1
3		0.3	0.1	< 0.1	<0.1	<0.1
4			0.1	< 0.1	<0.1	<0.1

Horizontal die Entfernung in Metern, auf die optimal fokussiert wurde; vertikal die Entfernung, in der gemessen wurde; in der Tabelle der scheinbare Winkel der Dicke der Laserlinie in Grad.

B Verwendete Software

Die auf dem autonomen System verwendete Software umfasst Sidux-Linux, ein Debian-Derivat; RP6-Lib, die mit dem Arexx-Conrad RP6 ausgelieferte C-Bibliothek für ATMega; C++ und den Gnu GCC Compiler; die OpenCV-Programmbibliothek; libSerial zur seriellen Kommunikation; den Webserver nginx.

Während der Entwicklung wurden verwendet: Ubuntu Linux; QCad Community Edition; Blender; Kile; Gimp; KBibTex; codelite; OpenOffice; Subversion; ein Unterprogramm der Software Australis zur Bestimmung von Passpunkten.

Der größere Teil dieser Software steht unter der *GNU General Public License / GPL* (Free Software Foundation, 2007) oder ähnlichen freien Lizenzen.

C Hardwarekomponenten

- Laser Huanic LH650-16-3-F. Wellenlänge 650 nm. Betriebsspannung 3-5 Volt. Leistungsaufnahme ≤ 120 mA. Lichtleistung 16 mW. Manuell fokussierbare Linienoptik mit 110° Streuung durch eine zylindrische Linse. Geräteklasse 2M nach EN 60825-1.
- CMOS-Farb-Kameramodul aus einer Webcam “Logitech QuickCam Web” (auch: “QuickCam Chat”). Manueller Fokus. Kontroll-Chip VIMICRO ZC-0302. “Blende” als reine Software-Lösung (Addition von Einzelbildern zur Verbesserung der Helligkeit). USB-Anschluss. Bildübertragung in JPEG-Kompression. Gesichtsfeld (experimentell ermittelt) $26.3 / 35.4^\circ$. Auflösungen 640 x 480 und 320 x 240. Bildrate (experimentell ermittelt): 14.26 Hz bei hoher, 28.56 Hz bei niedriger Auflösung. Die im Scanbetrieb erreichbare Bildrate beträgt nur $\frac{2}{3}$ davon; dies wird in Abschnitt 3.3.3 erklärt.
- Schrittmotor Nanotec STS2812S1006-B. Winkelauflösung im Halbschrittbetrieb: $\frac{2\pi}{400} = 0.9^\circ$. Betriebsspannung 2.66 V. Leistungsaufnahme 0.95 A. Schrittgenauigkeit 5%. Gewicht 242 g.
- Netbook “A110” des Retailers “One” (auch “Quanta IL1”). Technische Daten: CPU: VIA C7-M, 1 GHz; 2 GB Solid-State-Disk; 512 MB RAM; 7 Zoll TFT; WLAN-Modul Realtek RTL8187B. Anschlüsse: 2 USB 2.0; 1x LAN, VGA out. Laufzeit ca. 3 Stunden. Neupreis ca. 149 Euro.
- mobiles Robotersystem RP6, Hersteller: Arexx Engineering. Steuerung über Mikrocontroller Atmel ATmega32 mit Bootloader. Antrieb über 2 Gummiraupen; Radstand ca. 11.2 cm, Spurweite ca. 10.2, Breite der Raupen 1.6 cm. Konnektierbar und programmierbar über UART. Neupreis ca 129 Euro

D Betriebsanleitung ARP-117

Sicherheitshinweise

- Sicherstellen, dass niemand in den Laser blickt.
- Einschalten des RP6 bei angeschlossenem Ladegerät kann das Gerät beschädigen!
- Zum Stoppen des RP6 dessen Hauptschalter herunterdrücken. Das Gerät kann nicht ohne weiteres mit der Hand aufgehalten werden, da ein höherer Widerstand zu einer Nachregelung der Motorleistung führt!
- Der freie Speicherplatz des PC ist auf ca. 250 MB begrenzt. Sollen STL-Dateien oder Kamerabilder gespeichert werden, muss darauf geachtet werden, diesen nicht völlig aufzubauchen und so das Betriebssystem einzufrieren! Es muss in diesem Fall von einem USB-Stick gebootet werden um wieder Speicherplatz freizumachen.

Betrieb

1. Ladekabel vom RP6 trennen.
2. Ladekabel vom PC trennen.
3. PC einschalten
4. SSH-Verbindung herstellen: 192.168.1.4 (Ethernet) oder .14 (Wireless). SSID: ARP (kein Ad-Hoc-Modus verfügbar!). Subnetz 255.255.255.0. User: root ,
Passwort: laser117
5. Webbrowser mit der gleichen Adresse öffnen. Werden keine Bilder geladen, den Webserver nginx starten.
6. Schalter am UART-Kabel auf Position **X** stellen (Position **P** erlaubt Zugang zum RP6-Bootloader.)
7. Webcam und USB/UART-Wandler anschließen

8. RP6 einschalten
9. RP6 Starttaste betätigen
10. das auszuführende Programm ist “/home/arp117/ARP/Scanner”. Optionen:
Scanner [320|640] [calibration | fileset <Verzeichnis> | scan [drive]] ...

scan

drive

save speichert Einzelbilder jedes Scans - begrenzter Speicherplatz!

stl speichert STL-Datei jedes Scans - begrenzter Speicherplatz!

fileset <Pfad> lädt den Bildersatz eines Scans, ausschließlich mit der Option scan.

nosubpixel schaltet standardmäßig aktivierte Optionen zur Bildverbesserung ab

nonoise "

nodespeckle "

nocutoff "

freeze unterbindet Geradeaus- bzw. Drehbewegungen bei Option “drive”

straight "

turn "

display visualisiert die Scan-Ergebnisse unter X-Windows, nicht per SSH

640 höhere Scan-Auflösung, sehr langsam

320 niedrige Scan-Auflösung, Standardeinstellung, ggfs. skalieren

calibration Hilfsroutine zum Aufzeichnen der Bilder zur Ebenenkalibrierung

11. Stoppen des RP6 durch Herunterdrücken des Hauptschalters
12. Steuerprogramm durch **STRG-C** beenden
13. Logdateien befinden sich im datumscodierten Verzeichnis
14. Der PC kann durch kurzes Drücken des Hauptschalters heruntergefahren werden.

Literaturverzeichnis

- [Alexander u. a. 2007] ALEXANDER, R. D. ; HALL-MAY, M. ; KELLY, T. P.:
Certification of Autonomous Systems under UK Military Safety Standards. In:
Proceedings of the 25th International System Safety Conference (ISSC '07), 2007
- [Beraldin u. a. 1995] BERALDIN ; EL-HAKIM ; BLAIS: A Comparative Evaluation of
the Performance of Passive and Active 3-D Vision Systems. In: ? (1995)
- [Berger und Rumpe 2008] BERGER, Christian ; RUMPE, Bernhard: Autonomes
Fahren - Erkenntnisse aus der DARPA Urban Challenge. In: *it - Information
Technology* 50 (2008), Nr. 4, S. 258–264
- [Besl 1988] BESL: Active, Optical Range Imaging Sensors. In: *Machine Vision and
Applications* Vol. 1 (1988), S. 127–152
- [Blais 2004] BLAIS: Review of 20 years of range sensor development. (2004)
- [Bradsky und Kaehler 2004] BRADSKY, Gary ; KAEHLER, Adrian: *Learning OpenCV*.
O'Reilly Media, 2004
- [Canny 1986] CANNY, J.: A computational approach to edge detection. In: *IEEE
Trans. Pattern Anal. Mach. Intell.* 8 (1986), November, Nr. 6, S. 679–698
- [Chen und Kak 1987] CHEN, C. ; KAK, A.: Modeling and calibration of a structured
light scanner for 3-D robot vision. In: *IEEE International Conference on Robotics
and Automation* Bd. 4, März 1987, S. 807–815

- [Davis und Chen 2001] DAVIS, James ; CHEN, Xing: A laser range scanner designed for minimum calibration complexity. In: *Third International Conference on 3D Digital Imaging and Modeling*, 2001, S. 91–98
- [Edel 2009] EDEL, Stephan: *Automatische Farbkalibrierung durch statistische Auswertung geometrischer Merkmale in der RoboCup Standard Platform League*, Humboldt-Universität zu Berlin, Studienarbeit, 2009
- [Forest u. a. 2004] FOREST, Josep ; SALVI, Joaquim ; CABRUJA, Enric ; POUS, Carles: Laser stripe peak detector for 3D scanners. A FIR filter approach. In: *Proceedings of the 17th International Conference on Pattern Recognition (ICPR'04)* (2004)
- [Free Software Foundation 2007] FREE SOFTWARE FOUNDATION: *GNU General Public License*. 2007. – URL <http://www.gnu.org/licenses/gpl.html>
- [Haberjahn 2006] HABERJAHN, Mathias: *Umsetzung eines vollautomatisierten Objekterfassungs-Systems über Methoden phasengestützter Streifenprojektion und Photogrammetrie*, Brandenburgische Technische Universität Cottbus, Diplomarbeit, 2006
- [Harris und Stephens 1988] HARRIS, C. ; STEPHENS, M.: A Combined Corner and Edge Detector. In: *Proceedings of the 4th Alvey Vision Conference*, 1988, S. 147–151
- [Hirschmüller 2008] HIRSCHMÜLLER: Stereo Processing by Semi-Global Matching and Mutual Information Problem. (2008)
- [Holz u. a. 2008] HOLZ, Dirk ; LÖRKEN, Christopher ; SURMANN, Hartmut: Continuous 3D Sensing for Navigation and SLAM in Cluttered and Dynamic Environments. In: *Proceedings of the International Conference on Information Fusion (FUSION)*. Cologne, Germany, June/July 2008, S. 1469–1475
- [Kälviäinen u. a. 1994] KÄLVIÄINEN, Heikki ; HIRVONEN, Petri ; XU, Lei ; OJA, Erkki: Comparisons of Probabilistic and Non-probabilistic Hough Transforms. In: *Proc. of 3rd European Conference on Computer Vision ECCV'94*, 1994, S. 351–360

- [Konolige u. a. 2006] KONOLIGE, Kurt ; AGRAWAL, Motilal ; BOLLES, Robert C. ; COWAN, Cregg ; FISCHLER, Martin A. ; GERKEY, Brian P.: Outdoor Mapping and Navigation Using Stereo Vision. In: *ISER*, 2006, S. 179–190
- [Lemke 2009] LEMKE, Marcel: *Bestimmung von Orientierungsparametern optischer Systeme auf mobilen Plattformen*, Humboldt-Universität zu Berlin, Diplomarbeit, 2009
- [Luhmann 2003] LUHMANN, Thomas: *Nahbereichsphotogrammetrie*. Wichmann Verlag, 2003
- [Meffert und Hochmuth 2004] MEFFERT, Beate ; HOCHMUTH, Rolf: *Werkzeuge der Signalverarbeitung*. Pearson, 2004
- [Nelder und Mead 1965] NELDER, J. A. ; MEAD, R.: A simplex method for function minimization. In: *Computer Journal*, 1965, S. 308–313
- [Nicklin u. a. 2007] NICKLIN, Steven P. u. a.: Rolling Shutter Image Compensation. In: *Robocup 2006: Robot Soccer World Cup X: No. 10*, Springer, 2007 (Lecture Notes on Computer Science)
- [Pito 1995] PITO, Richard: Characterization, Calibration, and Use of the Perceptron Laser Range Finder in a Controlled Environment / University of Pennsylvania. 1995. – Forschungsbericht
- [Rocchini u. a. 2001] ROCCHINI, C. ; CIGNONI, P. ; MONTANI, C. ; PINGI, P. ; SCOPIGNO, R.: *A low cost 3D scanner based on structured light*. 2001
- [Schempf u. a. 2003] SCHEMPF, Hagen ; CROWLEY, William ; GASIOR, Chris ; MOREAU, David: Ultra-rugged soldier–robot for urban conflict missions. In: *Systems International (AUVSI) 30th Annu. Symp. and Exhibition*, 2003
- [SICK AG 2010] SICK AG: *Lasermesstechnik: LMS1xx / LMS151 / Outdoor / Short-Range*. 2010. – URL <https://www.mysick.com/partnerPortal/ProductCatalog/DataSheet.aspx?ProductID=33754>

- [Stiller u. a. 2008] STILLER, Christoph ; KAMMEL, Sören ; LULCHEVA, Irina ; ZIEGLER, Julius: Probabilistische Methoden in der Umfeldwahrnehmung Kognitiver Automobile. In: *at - Automatisierungstechnik* 56 (2008), November, Nr. 11, S. 563–574
- [Thrun u. a. 2006] THRUN, Sebastian ; BURGARD, Wolfram ; FOX, Dieter: *Probabilistic Robotics*. The MIT Press, 2006
- [Thrun u. a. 2003] THRUN, Sebastian ; HÄHNEL, D. ; FERGUSON, D. ; MONTEMERLO, M. ; TRIEBEL, R. ; BURGARD, Wolfram ; BAKER, C. ; OMOHUNDRO, Z. ; THAYER, S. ; WHITTAKER, W.: A System for Volumetric Robotic Mapping of Abandoned Mines. In: *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2003
- [Trucco und Verri 1998] TRUCCO, Emanuele ; VERRI, Alessandro: *Introductory Techniques for 3-D Computer Vision*. Prentice Hall, 1998
- [Winkelbach u. a. 2006] WINKELBACH, Simon ; MOLKENSTRUCK, Sven ; WAHL, Friedrich M.: Low-Cost Laser Range Scanner and Fast Surface Registration Approach. In: *Lecture Notes in Computer Science : Pattern Recognition* (2006), S. 718–728
- [Zhang 2000] ZHANG, Zhengyou: A Flexible New Technique for Camera Calibration. In: *IEEE Trans. Pattern Anal. Mach. Intell.* 22 (2000), Nr. 11, S. 1330–1334

Inhaltsverzeichnis des Datenträgers

```
|-- Ausarbeitung.pdf
|-- Code_Kalibrierung
|   |-- _homogen.cpp
|   |-- _homogen.h
|   '-- geometry.cpp  (Hauptprogramm zur Kalibrierung)
|-- Code_Steuerung  (des autonomen Systems)
|   |-- Mikrocontroller
|   |   |-- RP6Lib  (modifizierte RP6-Bibliothek)
|   |   |   ...
|   |   |-- drive1.c  (Hauptprogramm Steuerung MC-seitig)
|   |   '-- makefile
|   '-- PC
|       |-- Scanner.mk
|       |-- Scanner.project
|       |-- arpFilter.cpp
|       |-- arpFilter.h
|       |-- arpMatch.cpp
|       |-- arpMatch.h
|       |-- arpVisual.cpp
|       |-- arpVisual.h
|       |-- arp_common.h
|       |-- arp_file.cpp
```

```

|      |-- arp_file.h
|      |-- arpmap.cpp
|      |-- arpmap.h
|      |-- arpserial.cpp
|      |-- arpserial.h
|      |-- arptools.cpp
|      |-- arptools.h
|      |-- scanner.cpp    (Hauptprogramm Steuerung PC-seitig)
|      '-- scanner.h
|-- Experimente
|   |-- Autonome Fahrt 1    (Logdateien)
|   |   ...
|   |-- Autonome Fahrt 2    (Logdateien)
|   |   ...
|   |-- Nahbereichsmessung
|   |   |-- logARP_0426_215451    (Bildserie zur Reproduktion der Messung)
|   |   |   ...
|   |   |-- logARP_0426_222727    (Bildserie zur Reproduktion der Messung)
|   |   |
|   |   |-- main.cpp    (Hilfsprogramm zur Auswertung)
|   |   '-- neue_punkte.txt    (manuell selektierte Punkte)
|   |-- Weitbereichsmessung
|   |   |-- ob1q-320    (Bildserie zur Reproduktion der Messung)
|   |   |   ...
|   |   '-- ob1q-640    (Bildserie zur Reproduktion der Messung)
|   |
|   '-- simplex    (Nelder-Mead-Versuch)
|       '-- main.cpp
|-- Kalibrierung_verschiedenes

```

```

| |-- Laserfokussierung00.odt
| |-- contrast.c
| |-- innere Orientierung    (Messdaten aus Australis)
| |   |-- main.cpp
| |   |-- pointData.cpp    (Hilfsprogramm n. Lemke)
| |   '-- pointData.h
| |-- kalibrierungsdaten    (Beispieldatensatz zur relativen Orientierung)
| |
| |-- schach8x6+1.png    (Kalibrierungsmuster)
| |-- schach8x6+1.txt
| |-- testbild1.pdf
'-- Mechanik
    |-- Blende.dxf
    |-- Hilfsplatte_Frontscanner00.dxf
    |-- Kamerahalter-24.blend
    |-- Kamerakopf-Anordnung00.dxf
    |-- Liste_Montagematerial.txt
    |-- Rotor-017.blend
    '-- Untere_Aufbauplatte08-ermittelter_schwerpunkt.dxf

```

Danksagung

Für ihre Unterstützung während der Anfertigung dieser Arbeit möchte ich mich herzlich bedanken:

- bei meinem Vater, Peter Edel
- bei Prof. Ralf Reulke, Manfred Hoffmeister, Bernd Fiol, Sascha Bauer, Andreas Luber, und den Mitarbeitern des Instituts für Verkehrssystemtechnik am DLR Berlin-Adlershof
- bei Dr. Manfred Hild und Carsten Siedel, Humboldt-Universität zu Berlin, Arbeitsgruppe für künstliche Intelligenz
- bei Holger Heissmeyer, Bernd Volkert und Birgit Lulay