



Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Bachelor Thesis

Maik Pätzold

Entwicklung eines Kleinroboters für die
Kleinteilsuche

Maik Pätzold

Entwicklung eines Kleinroboters für die Kleinteilsuche

Bachelor Thesis eingereicht im Rahmen der Bachelorprüfung

im Studiengang Technische Informatik
am Department Informatik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer : Prof. Dr.-Ing. Andreas Meisel
Zweitgutachter : Prof. Dr. Reinhard Baran

Abgegeben am 17.11.2014

Maik Pätzold

Thema der Bachelorthesis

Entwicklung eines Kleinroboters für die Kleinteilsuche

Stichworte

Kleinroboter, Sensorik, Indoor Positionsbestimmung, visuelle Objekterkennung

Kurzzusammenfassung

Diese Arbeit beschreibt den Entwurf eines Kleinroboters mit den erforderlichen Sensoren für einen autonomen Betrieb. Weiterhin wird ein visuelles Verfahren vorgeschlagen und analysiert, dass eine absolute Positionsbestimmung nach einem Nutzereingriff ermöglicht. Abschließend werden für die Kleinteilsuche visuelle Verfahren zur Objekterkennung auf Basis von kaskadierten Klassifikatoren untersucht.

Maik Pätzold

Title of the paper

Development of a small Robot for small part search

Keywords

small robot, sensors, indoor location determination, visual object detection

Abstract

This bachelor thesis describes the conception of a small robot with necessary sensors for an autonomous operation. Furthermore a visual procedure is suggested and analysed that enables an absolute location determination after a user's intervention. Finally visual procedures for object recognition based on cascade classifiers are examined for the small part search.

Inhaltsverzeichnis

1	Einleitung	7
1.1	Ziel der Arbeit	7
2	Grundbegriffe	8
2.1	Simultaneous Localization and Mapping (SLAM).....	8
2.2	Kidnaped Robot Problem.....	8
2.3	RS-232	8
2.4	I2c	8
2.5	Transistor-Transistor-Logik Schaltung (TTL).....	8
3	Anforderungen	9
3.1.1	Hinderniserkennung	9
3.1.2	Treppendetektion	9
3.1.3	Nutzereingriff / Kidnapped Robot Problem	10
3.1.4	Positionsbestimmung / Lokalisation	10
3.1.5	Objekterkennung	10
3.1.6	Akkustandsüberwachung.....	10
3.1.7	Autarke Funktionsweise.....	11
3.1.8	Hindernis überfahren.....	11
3.1.9	WLAN Anbindung.....	11
4	Grundlagen.....	12
4.1	Odometrie.....	12

4.2	Hinderniserkennung	14
4.2.1	Ultraschallsensoren	14
4.2.2	Infrarotsensoren	15
4.3	Lagebestimmung.....	16
4.3.1	Gyroskop	16
4.3.2	Accelerometer	19
4.3.3	Skalierung und Offset.....	20
4.3.4	Integration	21
4.3.5	Komplementärfilter	21
4.4	Kompass.....	23
4.4.1	Offset	23
4.5	Pulsweiten-Modulations-Verfahren (PWM)	24
5	Entwurf des Prototypen	26
5.1	Chassis	26
5.2	Architektur	27
5.3	Systemmodul	28
5.4	IO-Modul.....	29
5.5	Sensoren	30
5.5.1	Hinderniserkennung	30
5.5.2	Treppendetektion	31
5.5.3	Temperatursensor	33
5.5.4	Lagesensor	34
5.5.5	Kompass.....	34
5.5.6	Odometrie.....	34
5.5.7	Sensormodul	35
5.5.8	Heckmodul.....	35
5.5.9	Frontmodul	36
5.6	Antrieb	36
5.7	Spannungsversorgung.....	37
5.7.1	LiPo Akku.....	37
5.7.2	5V Versorgungsspannung	38

5.8	Kameras	39
5.9	Software.....	39
5.9.1	Front-/Heckmodul.....	40
5.9.2	Senormodul.....	41
5.9.3	IO-Modul.....	44
5.9.4	Sensormodul.....	45
5.10	Zusammenfassung und Kosten	47
6	Positionsbestimmung	49
6.1	Odometrie Referenzpunktkorrektur	49
6.2	Positionsbestimmung mittels Landmarke.....	52
6.2.1	Farbsegmentierung anhand des HSV Farbraums.....	52
6.2.2	Sobel	53
6.2.3	Kantenverfolgung / Strukturanalyse	55
6.2.4	Erkennen der Landmarke und ihres Rotationswinkels	55
6.2.5	Positionsbestimmung	56
6.2.6	Tests.....	58
6.2.7	Fazit.....	60
7	Objekterkennung.....	61
7.1.1	Haar-like Features	61
7.1.2	Local Binary Patterns (LBP)	64
7.1.3	Histograms of oriented Gradients (HOG).....	65
7.1.4	Cascade-Classifier	66
7.1.5	Sliding Window Verfahren	69
7.1.6	Training	70
7.1.7	Vergleich	70
7.1.8	Fazit.....	72
8	Zusammenfassung und Ausblick	73
8.1	Zusammenfassung	73
8.2	Ausblick.....	73
9	Anhang	82

1 Einleitung

Gerade in kreativen jungen Unternehmen gehören Spielzeuge zur Firmenkultur. Sie fördern die Teambildung und sind eine willkommene Abwechslung im oft stressigen und an Deadlines ausgerichteten Alltag. Neben Tischfußball oder Tischtennis erfreuen sich sogenannte Nerf Guns immer größerer Beliebtheit. Dies sind Spielzeuggewehre, mit denen Munition in Form von Schaumstoff-Pfeilen oder -Scheiben abgeschossen werden kann. Diese Munition fliegt beim Spielen in alle Richtungen und muss danach einzeln eingesammelt werden, damit die Nerf Guns erneut befüllt werden können. Um diese mühsame und zeitraubende Arbeit zu umgehen, entstand die Idee, einen Roboter zu entwickeln, der in der Lage ist, diese Munition selbstständig zu suchen, einzusammeln und an einer Station abzuladen.

1.1 Ziel der Arbeit

Im Rahmen dieser Arbeit soll ein fahrtüchtiger Prototyp eines Kleinroboters entstehen. Dieser soll im Nachgang als Grundlage für die Entwicklung des zuvor beschriebenen Sammel-Roboters dienen. Hierbei soll eine geeignete Hardware-Plattform mit den erforderlichen Sensoren entworfen werden, die eine spätere Entwicklung des autarken und autonomen Sammel-Roboters ermöglicht. Die erforderliche Mechanik zum Einsammeln der Munition soll im Rahmen dieser Arbeit jedoch nicht berücksichtigt werden. Der zu entwerfende Prototyp soll ein Budget von 300,- € nicht übersteigen. Weiterhin soll im Rahmen dieser Arbeit ein Verfahren zur Positionsbestimmung vorgeschlagen und analysiert werden, das dem Roboter eine Lokalisierung auch nach einem Nutzereingriff ermöglicht. Abschließend sollen Methoden zur visuellen Erkennung der Munition auf Basis von kaskadierten Klassifikatoren untersucht und bewertet werden.

2 Grundbegriffe

2.1 Simultaneous Localization and Mapping (SLAM)

Beim SLAM handelt es sich um ein Verfahren zur gleichzeitigen Positionsbestimmung und Kartographie. Das SLAM Verfahren ist meist eine Lösung für Roboter, bei welchen eine direkte Positionsbestimmung nicht möglich ist. Hierbei wird versucht gleichzeitig sowohl die eigene Position, als auch eine Umgebungskarte auf Basis von Sensordaten inkrementell zu bestimmen und im weiteren Verlauf anhand der Kartendaten die eigene Position ggf. zu korrigieren.

2.2 Kidnaped Robot Problem

Das Kidnaped Robot Problem tritt auf, wenn durch einen externen Eingriff eine durch den Roboter inkrementell bestimmte Position ohne Wissen des Roboters ungültig wird. So geht der Roboter noch von einer Position aus an der er sich durch den Eingriff tatsächlich nicht mehr befindet.

2.3 RS-232

Bei RS-232 handelt es sich um ein serielles Kommunikationsprotokoll zwischen zwei Endgeräten.

2.4 I2c

Beim I2C handelt es sich um ein Master-Slave Bus System zur Kommunikation zwischen einem Master und mehreren Slave Endgeräten. Der Master ist hierbei die Verwaltende Instanz und leitet die Kommunikation ein. Ein Slave ist ein Endgerät das durch den Master gesteuert bzw. abgefragt wird.

2.5 Transistor-Transistor-Logik Schaltung (TTL)

TTL ist eine Schaltungstechnik für die direkte logische Verknüpfung von Schaltungselementen.

3 Anforderungen

Die Anforderungen an die Hardware des Prototypen und Verfahren zur Lokalisation sowie Objekterkennung sind folgende:

- Erkennen von Hindernissen zur Kollisionsvermeidung
- Erkennen von Treppen zur Vermeidung von Abstürzen
- Erkennen von Nutzereingriffen zur automatischen Abschaltung und erneuter Positionsbestimmung
- Positionsbestimmung ohne aufwendige externe Ortungssysteme
- Visuelles Erkennen der Munition
- Überwachung des Akku-Ladezustandes
- Autarke Funktionsweise
- Möglichkeit kleine Hindernisse überfahren zu können
- WLAN Anbindung
- Maximale Kosten für den Prototypen von 300,- €

Die Mechanik zum Einsammeln der Munition soll im Rahmen dieser Arbeit jedoch keine Rolle spielen. Im folgendem werden die zuvor gestellten Anforderungen kurz erläutert.

3.1.1 Hinderniserkennung

Um einen autonomen Betrieb des Roboters zu ermöglichen, muss dieser in der Lage sein, Hindernisse zu erkennen. Dies dient primär der Kollisionsvermeidung und kann im Nachgang dieser Arbeit auch für die Erstellung einer Hinderniskarte dienen. Diese Hinderniskarte kann wiederum für eine gezielte Navigation im Raum und so auch für eine systematische Suche dienen. Hierfür sind Sensoren notwendig, die sowohl bei Vorwärts- als auch Rückwärtsfahrten Hindernisse erkennen können. Auch sollen Hindernisse bei Seitwärtsbewegungen und Drehungen erkannt und somit Kollisionen vermieden werden.

3.1.2 Treppendetektion

Um einen Absturz des Roboters, beispielweise aufgrund einer im Raum befindlichen Treppe, nach unten zu vermeiden, muss der Roboter über Sensoren verfügen, die

den Abstand zum Boden messen. Dadurch sollen Öffnungen im Boden erkannt und Abstürze vermieden werden. Die Sensoren sollen so gewählt werden, dass sowohl bei Vorwärts- als auch Rückwärtsfahrten Öffnungen erkannt werden können.

3.1.3 Nutzereingriff / Kidnapped Robot Problem

Da der Roboter jederzeit seine Position bestimmen können soll, muss dieser in der Lage sein, einen Nutzereingriff zu erkennen und so ggf. eine Korrektur der aktuellen Position vorzunehmen. Dies wird auch als das Kidnapped Robot Problem bezeichnet. Auch soll die Erkennung eines Nutzereingriffes dazu genutzt werden, den Antrieb des Roboters abzuschalten. Dadurch sollen sowohl Verletzungen des Nutzers als auch Beschädigungen des Roboters vermieden werden.

3.1.4 Positionsbestimmung / Lokalisation

Die Genauigkeit der Positionsbestimmung ist hierbei wichtig, um im Nachgang dieser Arbeit eine Ablade- oder Ladestation anfahren zu können. Weiterhin ist die Positionsbestimmung entscheidend für die Genauigkeit bei der Erstellung von Umgebungskarten, um mit diesen ein systematisches Durchsuchen von Räumen zu ermöglichen. Im Rahmen dieser Arbeit ist jedoch die wichtigste Anforderung an die Positionsbestimmung die Lösung des Kidnapped Robot Problems. Dies soll gewährleisten, dass der Roboter auch nach einem Nutzereingriff in der Lage ist, seine Position im Raum zu bestimmen. Um die Kosten so gering wie möglich zu halten und auch die Inbetriebnahme so einfach wie möglich zu gestalten, soll auf den Einsatz von externen Ortungssystemen verzichtet werden. Daher wird ein Verfahren vorgeschlagen und analysiert, dass durch visuelle Landmarkenerkennung eine Lösung des Kidnapped Robot Problems ermöglicht.

3.1.5 Objekterkennung

Um die Munition zu finden, soll auf Verfahren der visuellen Objekterkennung zurückgegriffen werden. Hierzu soll die frei verfügbare Computer Vision Bibliothek opencv eingesetzt und die möglichen Verfahren auf ihre Leistungsfähigkeit untersucht werden.

3.1.6 Akkustandsüberwachung

Zum Schutz des Akkus soll der Roboter in der Lage sein, den Ladezustand des Akkus überwachen zu können, um ggf. alle Systeme abzuschalten. Ein weiterer Nutzen

dieser Ladestandskontrolle ist die Möglichkeit, im Nachgang dieser Arbeit ggf. automatisch eine Ladestation anzufahren, sobald dies erforderlich wird.

3.1.7 Autarke Funktionsweise

Der Roboter soll ohne externe Recheneinheit auskommen. Das heißt, alle erforderlichen Verfahren sollen direkt auf der Hardware des Roboters ablaufen.

3.1.8 Hindernis überfahren

Bei der Auswahl des Chassis und des Antriebs des Prototypen soll berücksichtigt werden, dass kleinere Hindernisse überfahren werden können.

3.1.9 WLAN Anbindung

Für den Zugriff auf den Prototypen soll dieser über eine WLAN Anbindung verfügen. Dieser kann später auch für eine Telemetrieübertragung genutzt werden.

4 Grundlagen

4.1 Odometrie

Die am weitesten verbreitete und einfachste Methode zur indirekten Positionsbestimmung ist die Odometrie. Hierbei wird der zurückgelegte Weg direkt anhand der Radumdrehung und dem bekannten Radumfang bestimmt. In Kleinrobotersystemen dienen zumeist eine Infrarot Reflex- oder Gabellichtschranke und eine Encoder-Scheibe am Antrieb bzw. Getriebe als Odometrie-Sensor. Hierbei sind die Verfahren in Inkremental- und Absolutdrehgeber unterteilt. Beim Inkrementaldrehgeber wird eine Encoder-Scheibe in helle und dunkle Flächen unterteilt, wie in Abbildung 1 dargestellt. Eine unmittelbar vor dieser Encoder-Scheibe positionierte Reflexlichtschranke erzeugt nun aufgrund der unterschiedlich reflektierenden Abschnitte ein Rechtecksignal. Dieses Rechtecksignal kann als Takt für einen Zähler genutzt werden, welcher bei fallender und/oder steigender Flanke zählt. Ein absoluter Drehgeber ist zusätzlich in der Lage, die aktuelle Drehposition eindeutig anzugeben. Dazu ist eine, beispielweise mittels mehrerer Spuren (siehe Abbildung 2), binärkodierte Encoder-Scheibe angebracht.

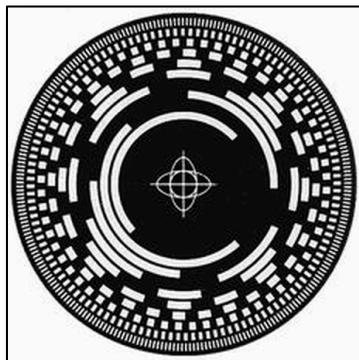


Abbildung 2 absolut Encoder-Scheibe
(Ohne, elcis-encoder.com)

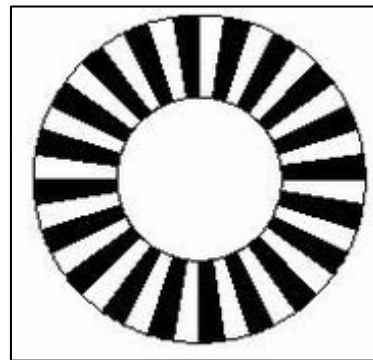


Abbildung 1 inkremental Encoder-Scheibe
(Ohne, Drehgeber)

Jede dieser Spuren wird wie beim Inkrementaldrehgeber in ein Rechtecksignal umgesetzt. Aus der Kombination der einzelnen Rechtecksignale ergibt sich somit eine eindeutig binärkodierte Position. Ein weiterer Vorteil dieses Verfahrens ist, dass auch die Drehrichtung aus den Zustandsfolgen abgelesen werden kann. Wird statt einer Reflex- eine Gabellichtschranke genutzt, sind die Encoder-Scheiben entsprechend in lichtdurch- und lichtundurchlässige Abschnitte unterteilt. Dabei

unterscheiden sich die Reflex- und Gabellichtschrankensysteme nur geringfügig in ihrem Konstruktionsaufwand und ihrer Empfindlichkeit gegenüber Fremdlicht. Bei Antrieben basierend auf schritt- oder bürstenlosen Motoren (Synchronmaschine) lässt sich die zurückgelegte Distanz und Drehposition auch direkt aus den Daten der Motorsteuerung gewinnen. Bei einem Differentialantrieb kann die Position mittels Odometrie wie folgt inkrementell berechnet werden (Poschmann, 2009) (Ohne, Absolutwertgeber) (Ohne, Inkrementalgeber) (Ohne, elcis-encoder.com):

$$\delta = \frac{\Delta L + \Delta R}{2} \quad (1)$$

$$\alpha = \frac{\Delta L - \Delta R}{b} \quad (2)$$

Wobei ΔR und ΔL die zurückgelegte Strecke mit dem rechten bzw. linken Rad und b den Radabstand angeben. (Poschmann, 2009) Die zurückgelegte Distanz jeder Antriebsseite lässt sich aus den Takten der verwendeten Odometriesensoren mit folgender Formel bestimmen:

$$\Delta s = \frac{2 * \pi * r}{m} * \Delta k \quad (3)$$

Hierbei gibt Δs die Distanz in mm, Δk die gezählten Takte, r den Radius des Antriebsrades und m die Anzahl Takte für eine volle Radumdrehung an. So ergibt sich aus (1) die mittlere zurückgelegte Strecke und aus (2) der Kurs. Um hieraus eine Position inkrementell bestimmen zu können, ist die vorherige Position und der vorherige Kurs notwendig. So kann die Position anhand folgender Formel berechnet werden (Poschmann, 2009):

$$x_t = \begin{pmatrix} x_{t-1} + \delta * \cos \theta \\ y_{t-1} + \delta * \sin \theta \\ \theta_{t-1} + \alpha \end{pmatrix} \quad (4)$$

Aufgrund folgender Faktoren kommt es bei der inkrementellen Positionsbestimmung mit Odometrie zu Ungenauigkeiten:

1. Systematische Fehleranteile
 - Radgeometrie (Unrundheit, Verschleiß, ungenaue Bestimmung des Raddurchmessers)
 - Fahrgestellgeometrie (Spiel, ungenaue Bestimmung der Radabstände und Lenkeinrichtungen)
 - Fahrzeuggewicht (ungleiche Verteilung: stärkere Belastung und Verformung einzelner Räder)
2. Bodenbeschaffenheit (Unebenheit, Schlupf)
3. Wind

Die systematischen Fehleranteile lassen sich beispielsweise durch eine Kalibrierung bestimmen und reduzieren. Der sich aufsummierende Fehler, auch Schleppfehler genannt, lässt sich somit lediglich minimieren. (Poschmann, 2009)

4.2 Hinderniserkennung

4.2.1 Ultraschallsensoren

Beim Ultraschallsensor handelt es sich um einen akustischen Sensor zur Abstandsmessung. Hierzu wird ein Ultraschallsignal, welches ab einer Frequenz von ca. 16KHz beginnt, ausgesendet und eine Laufzeitmessung des eintreffenden Echos vorgenommen. Zur Minimierung von Fremdeinwirkungen durch andere Schallquellen werden die Signale in der Regel speziell moduliert. Handelsübliche Ultraschallsensoren senden meist ein gepulstes Signal von ca. 40KHz aus. Aufgrund der relativ hohen Frequenz von Ultraschall sind sowohl Sender als auch Empfänger oftmals mittels piezoelektrischer Wandler realisiert. Ein Vorteil von Schall ist die relativ geringe Ausbreitungsgeschwindigkeit von ca. 343m/s bei 20°C in Luft. Diese erlaubt eine Laufzeitmessung mit relativ einfacher Hardware. Im Hinblick auf eine Kollisionserkennung mittels Ultraschall-sensoren ist der relativ große Öffnungswinkel des ausgehenden Signals ein Vorteil dieser Technik. Dieser erlaubt eine relativ großflächige Suche nach Hindernissen mit nur einem Sensor. Gleichzeitig ist dieser Öffnungswinkel im Hinblick auf die Erstellung von Landkarten problematisch, da so selbst kleinste Hindernisse nicht von großen Hindernissen unterschieden werden können.

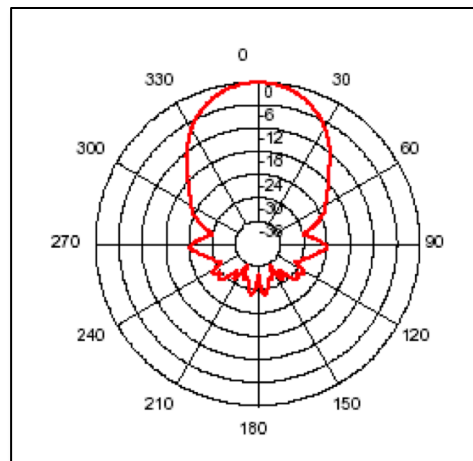


Abbildung 3 Signalform (Ohne, SRF05 Specs)

Das ausgehende Signal solcher Sensoren ist meist keulenförmig (siehe Abbildung 3) und besitzt einen Öffnungswinkel zwischen 15° und 30° . Aufgrund der Ablenkung bzw. der Unterdrückung ausgehender Schallwellen sind spitze Winkel zu Objekten und schallabsorbierende Materialien problematisch und daher zu vermeiden. Da Schallwellen relativ stark temperaturabhängig sind, ist eine Kompensation zu empfehlen. (Ohne, Ultraschall) (Ohne, SRF04 Specs) (Ohne, Ultraschallsensoren) Als Näherungsformel zur Berechnung der Schallgeschwindigkeit in Luft zwischen -35°C und $+35^\circ\text{C}$ abhängig von der Temperatur kann folgende Formel genutzt werden:

$$c_{\text{luft}} = (331,5 + 0,6 * \vartheta) \frac{\text{m}}{\text{s}} \quad (5)$$

Wobei ϑ der gemessenen Temperatur in $^\circ\text{C}$ entspricht. (Ohne, sengpielaudio.com) Daraus ergibt sich folgende Formel zur Berechnung der Entfernung eines Objektes anhand der Laufzeit und in Abhängigkeit der Temperatur:

$$d = \frac{(331,5 + 0,6 * \vartheta) * \Delta t}{2} \quad (6)$$

Hierbei entspricht Δt der gemessenen Laufzeit in Sekunden und ϑ der gemessenen Temperatur in $^\circ\text{C}$. Das Ergebnis d gibt somit die Entfernung in Meter an.

4.2.2 Infrarotsensoren

Wie die meisten Sensoren zur Entfernungsmessung lassen sich auch lichtbasierte Systeme mittels Laufzeitmessung und Trilateration oder Winkelmessung und

Triangulation realisieren. Jedoch ist eine Laufzeitmessung bei geringen Abständen aufgrund der hohen Geschwindigkeit von Licht sehr aufwändig. Daher wird bei handelsüblichen Infrarot-Distanzsensoren zumeist mittels Triangulation die Entfernung zu einem Objekt gemessen. Hierzu wird das Licht einer Infrarot-LED mit Hilfe einer Linse zu einem relativ schmalen Strahl gebündelt. Der reflektierte Lichtstrahl wird anschließend durch eine weitere Linse auf einen Lichtsensor umgelenkt (siehe Abbildung 4). Dieser besitzt auf seiner Oberfläche unterschiedlich empfindliche Bereiche und erzeugt daher eine von der Position des eintreffenden Lichtstrahls und somit auch der Entfernung zum Objekt abhängige Spannung. Hierbei gilt es ungünstige Winkel zu den Objekten und absorbierende oder durchlässige Materialien zu vermeiden. Da diese eine zuverlässige Abstandsmessung verhindern können. Um den Einfluss von Fremdlicht auf solche Sensoren zu minimieren, wird das ausgehende Licht des Sensors zumeist speziell moduliert. Eine Eigenheit solcher Sensoren ist die konstruktionsbedingte relativ enge Begrenzung der messbaren Entfernung. Diese ist sowohl was den minimalen als auch maximalen Abstand betrifft von der Dimension des Lichtsensors und seiner Lage abhängig. Daher sind solche Sensoren zumeist mit unterschiedlichen Messbereichen für die jeweiligen Einsatzzwecke verfügbar. (Ohne, roboticlab.eu)

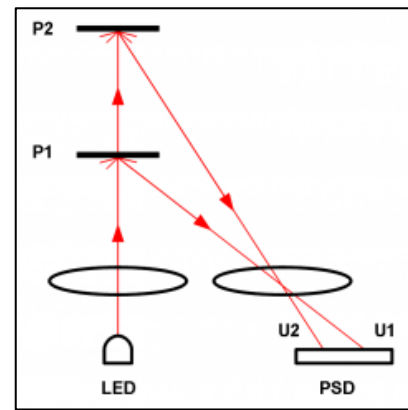


Abbildung 4 IR-Distanzmessung (Ohne, roboticlab.eu)

4.3 Lagebestimmung

Die Lagebestimmung des Roboters ist für die Erkennung von Nutzereingriffen vorgesehen. Im Nachgang dieser Arbeit kann die Lagebestimmung jedoch auch für die Korrektur von lageabhängigen Bildverzerrungen genutzt werden. Für die dreidimensionale Lagebestimmung werden hierbei ein Winkelbeschleunigungsmesser (Gyroskop) sowie ein Linearbeschleunigungsmesser (Accelerometer) eingesetzt. Diese sind einfach zu handhaben und kostengünstig zu beschaffen. So sind heutzutage Gyroskope und Accelerometer in fast jedem Smartphone verbaut und daher als Massenprodukt günstig herzustellen und gut verfügbar.

4.3.1 Gyroskop

Ein Gyroskop ist ein Instrument zur Messung der Winkelgeschwindigkeit, also der Drehgeschwindigkeit um eine Achse herum. Man spricht in diesem Zusammenhang

auch von einer rotatorischen Bewegung. Das Grundprinzip des Gyroskops geht auf die Erfindung des Kreiselkompasses von Johann Gottlieb Friedrich von Bohnenberger aus dem Jahr 1817 zurück. Dabei wird sich die Drehimpulserhaltung eines Kreisels zu nutze gemacht, aufgrund derer ein Kreisel extern einwirkenden Kräften ausweicht. Die dabei anfallende Corioliskraft kann gemessen werden und zeigt, je nach Aufbau des Kreiselinstrumentes, die rotatorischen Bewegungen der verschiedenen Achsen (siehe Abbildung 5). Diese werden typischerweise in Grad pro Sekunde ($^{\circ}/s$) angegeben. Auf Basis dieses Funktionsprinzips gingen unter

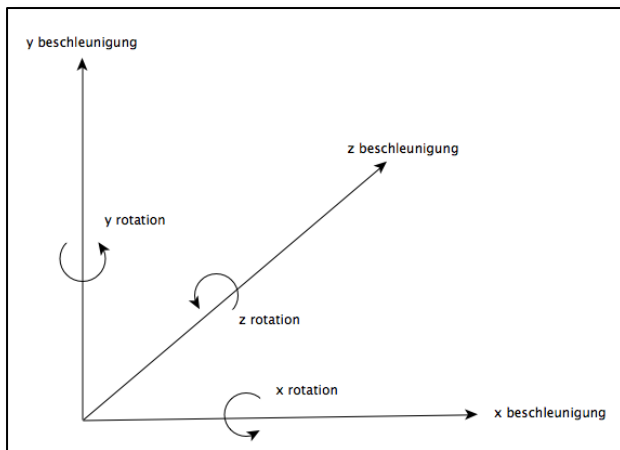


Abbildung 5 Beschleunigungsachsen

Gyroskope aufgrund mechanischer Unregelmäßigkeiten, welche herstellungsbedingt auftreten, temperaturabhängigen Veränderungen, Vibrationen, Stößen und elektrostatischen Effekten einem sogenannten Drift. Dieser Drift wird in Grad pro Stunde ($^{\circ}/h$) angegeben und setzt sich sowohl aus systematischen wie auch zufälligen Anteilen zusammen. Der systematische Fehler- bzw. Driftanteil lässt sich zwar durch geeignete mathematisch-physikalisch Modelle und Filter minimieren, jedoch nicht vollständig beseitigen. Aufgrund dieses sich inkrementell aufsummierenden Fehlers ist ein Gyroskop allein zur Lagebestimmung nicht praktikabel und bedarf weiterer Sensoren zur Korrekturmessung. Ein weiterer Grund, der die Lagebestimmung allein mit einem Gyroskop verhindert, ist die Tatsache, dass eine direkte Messung der Lage nicht möglich ist. Dies ist darin begründet, dass ein Gyroskop nur die momentan wirkende Drehrate angibt. (Ohne, Moderne Gyros und AHRS) (Ohne, reliablecomputing.eu, 2010) (Ohne, Kreiselinstrument)

Corioliskraft

Heutige vor allem im Consumer Bereich (z.B. Smartphones) eingesetzte elektronische Gyroskope sind sogenannte MEMS (Microelectromechanical System) Chips und arbeiten nach dem Vibrationskreiselpinzip. Bei diesem wird die Tatsache ausgenutzt, dass die Corioliskraft ein sich im Raum drehendes vibrierendes bzw.

anderem auch der künstliche Horizont, Wendezeiger und Kurskreisel, welche in jedem Flugzeugcockpit zu finden sind, hervor. Moderne elektronische Gyroskope basieren im Wesentlichen auf der Messung der Corioliskraft oder des Sagnac Effektes. Neben dem unterschiedlichen physikalischen Aufbau unterscheiden sich beide Systeme auch in Genauigkeit und Aufwand. So unterliegen

schwingendes System beeinflusst. Dabei steht der Grad der Beeinflussung im direkten Zusammenhang zur Drehrate. Zur Herstellung solcher Schwingungssysteme werden in MEMS basierten Gyroskopen unterschiedlichste geometrische Formen eingesetzt (z.B. Stimmgabel, schwingender Kamm, schwingender Ring) und die Beeinflussung des Systems meist mit kapazitiven Wandlern gemessen. Der Drift heutiger MEMS Gyroskope liegt im Bereich unter $10^\circ/\text{h}$. (Ohne, Moderne Gyros und AHRS)

Sagnac Effekt

Bei Gyroskopen auf Basis des Sagnac Effektes macht man sich zu nutzen, dass sich Licht konstant und unabhängig von der Bewegung des Bezugssystems ausbreitet. So ist Licht, das auf der sich drehenden Erde in den Raum emittiert wird, genauso schnell wie Licht, das im Raum emittiert wird. Das heißt die Geschwindigkeiten addieren sich nicht. Genau dieser Effekt macht den in Abbildung 6 gezeigten Aufbau möglich. Wenn sich nun der gesamte Aufbau im Uhrzeigersinn dreht, hat ein Photon, das sich ebenfalls im Uhrzeigersinn durch die Röhre zum nächsten Spiegel bewegt, einen längeren Weg als ein Photon, das sich gegen den Uhrzeigersinn bewegt. Diese Wegdifferenzen erzeugen zwei unterschiedliche Resonanzfrequenzen, deren Differenz proportional abhängig zur Drehrate des Gesamtaufbaues ist. Aufgrund der Tatsache, dass die verwendeten Spiegel nicht perfekt sein können, werden jedoch Teile des einen Lichtstrahls beim anderen eingekoppelt und es entsteht ein sogenannter Lock-In Effekt. Dieser bewirkt, dass sehr kleine Drehraten des Gesamtaufbaues nicht messbar sind. Das

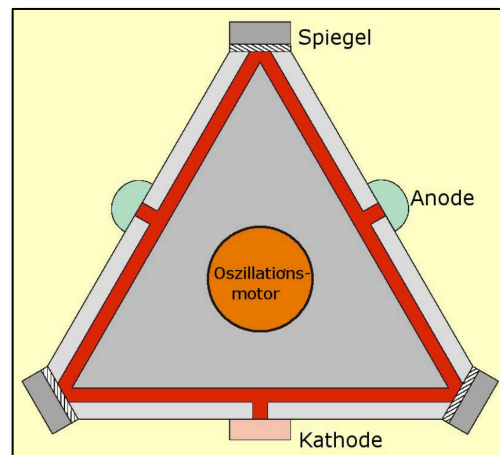


Abbildung 6 Ring-Laser Gyroskop Schema (Ohne, Moderne Gyros und AHRS)

Problem wird durch den in Abbildung 6 eingezeichneten Oszillationsmotor behoben. Der Motor versetzt den Gesamtaufbau in eine konstante leichte Bewegung, die am Ende wieder herausgerechnet wird. Neben dem hier gezeigten Ring-Laser Gyroskop gibt es auch noch das fiberoptische Gyroskop. Aufgrund seiner Ähnlichkeit zum Ring-Laser Gyroskop soll hier aber nicht weiter darauf eingegangen werden. Wie die MEMS Gyroskope unterliegen auch die optischen Systeme einem Drift. Dieser ist jedoch im Vergleich erheblich kleiner und bewegt sich für den Ring-Laser typischerweise im Bereich von unter $0,01^\circ/\text{h}$. (Ohne, Moderne Gyros und AHRS) (Ohne, Sagnac)

Fazit

Aufgrund des hohen Aufwandes und den damit verbundenen Kosten für Gyroskope, die mit dem Sagnac Effekt arbeiten, spielen diese im Consumer Bereich keine Rolle. Daher wird im Rahmen dieser Arbeit auch auf ein handelsübliches MEMS Gyroskop aus dem Consumer Bereich gesetzt. Diese sind sowohl einfach als auch günstig zu beschaffen sowie einfach zu handhaben.

4.3.2 Accelerometer

Im Gegensatz zu einem Gyroskop wird mit einem Accelerometer die lineare Beschleunigung entlang einer Achse gemessen (siehe Abbildung 5). Dies wird auch als translatorische Beschleunigung bezeichnet. Im Vergleich zum Gyroskop ist die Funktionsweise eines Accelerometers deutlich einfacher. So wird im einfachsten Fall die auf eine Testmasse wirkende Trägheitskraft gemessen. Da auch die Erdbeschleunigung auf eine solche Testmasse wirkt, ist auch diese ein Bestandteil der Messung. Eine weitere Möglichkeit der Beschleunigungsmessung ist der Ferraris-Sensor. Im folgendem werden diese beiden Möglichkeiten der Beschleunigungsmessung kurz vorgestellt.

Trägheitskraft

Die Messung der Trägheitskraft lässt sie sich im einfachsten Fall mit Hilfe einer Testmasse, welche am Ende einer Feder befestigt ist, nachvollziehen. Wird die Feder nun bewegt, führt die auf die Masse wirkende Trägheitskraft zu einer Dehnung der Feder. Auch die Erdbeschleunigung lässt sich so anhand der Stauchung der Feder nachvollziehen. Ein Nachteil dieser Methode ist jedoch das wirkende Zentrifugalkräfte zu einer Verfälschung der Messung führen können. Der Umstand einer nahezu konstanten Erdbeschleunigung erlaubt somit auch Rückschlüsse auf die Lage der Feder in Ruhe. Moderne Accelerometer werden wie Gyroskope als MEMS Chips gefertigt. Dabei werden nur wenige Mikrometer breite Stege mit Testmasse aus dem Silizium geätzt. Die trägheitsbedingte Auslenkung dieser Stege bewirkt dabei eine Änderung der elektrischen Kapazität zwischen sich selbst und einer festen Bezugselektrode. Diese Änderung wiederum kann gemessen und in eine Beschleunigung umgerechnet werden. Hierbei wird die Beschleunigung typischerweise in g angegeben, wobei 1g der mittleren Erdbeschleunigung entspricht. Wie Gyroskope unterliegen auch Accelerometer einem temperaturbedingten Drift. Dieser ist jedoch vergleichsweise klein und daher vernachlässigbar. (Ohne, Beschleunigungssensor) (Ohne, Moderne Gyors und AHRS)

Ferraris-Sensor

Der Ferraris-Sensor arbeitet ohne das zuvor beschriebene Feder-Masse System. Daher ist der Ferraris-Sensor auch unabhängig von Erdbeschleunigung und Zentrifugalkraft. Bei diesem System wird die relative Beschleunigung zwischen zwei Bauteilen gemessen. Dies geschieht mit Hilfe des von Galileo Ferraris entdeckten Wirbelstromprinzips. So werden durch das in einen Magneten (siehe Abbildung 7),

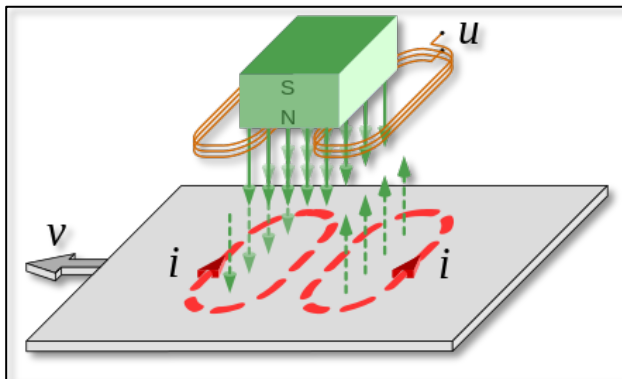


Abbildung 7 Ferraris-Sensor Schema (Ohne, Ferraris-Sensor)

im Bild grün dargestellt, erzeugte Magnetfeld im Beweglichen und Leitfähigen Teil darunter Wirbelströme erzeugt. Diese wiederum erzeugen sekundäre Magnetfelder, deren Stärke von der Geschwindigkeit abhängt. Aufgrund der sekundären Magnetfelder wird eine Spannung in der Abnehmerspule induziert (im Bild gelb dargestellt).

Diese Spannung wiederum kann gemessen werden und ist proportional zur Beschleunigung. Aufgrund der Unabhängigkeit von Erdbeschleunigung und Zentrifugalkraft bieten solche Sensoren eine vergleichsweise hohe Genauigkeit. Sie sind jedoch aufgrund der Relativmessung zwischen zwei Teilen im Rahmen dieser Arbeit nicht nutzbar. (Ohne, Ferraris-Sensor)

4.3.3 Skalierung und Offset

Im weiteren Verlauf des Kapitels Lagebestimmung geht es um die Integration der von MEMS basierter Gyroskope und Accelerometer gelieferten Sensordaten. Da es sich hierbei um digitale Sensoren handelt, ist der Wertebereich der Messdaten beschränkt. Zur Erhöhung der Genauigkeit verfügen diese Sensoren typischerweise über eine konfigurierbare Sensitivität. So lässt sich der Wertebereich bei einem Gyroskop auf eine maximale Winkelbeschleunigung pro Sekunde und der Wertebereich eines Accelerometer auf eine maximal g Beschleunigung skalieren. Zur Bestimmung der Winkelbeschleunigung bzw. der g Beschleunigung anhand der gelesenen Sensordaten muss diese Skalierung wieder rückgängig gemacht werden. Daher gilt:

$$x = \frac{\text{Sensitivität}}{\text{MAX}(x)} * x_{\text{gelesen}} \quad (7)$$

Weiterhin sind solche Sensoren sowohl herstellungsbedingt als auch durch ungenaue Montage mit einem Offset versehen. Der Offset ist hierbei die Verschiebung der Achse in Nulllage. Das heißt, die Messwerte der Achsen sind in Nulllage um einen konstanten Wert verschoben. Dieser Wert muss zuvor in Ruhe bestimmt werden und wird später bei jeder Messung abgezogen. (Ohne, all-electronics.de)

4.3.4 Integration

Wie zuvor gezeigt, ist nur der Einsatz eines Gyroskops zur Lagebestimmung nicht praktikabel. So wäre es lediglich möglich durch Aufsummieren eine relative Lage seit Beginn der Messung zu berechnen. Diese würde jedoch aufgrund des Drifts und den sich dadurch aufsummierenden Fehlern mit zunehmender Zeit immer ungenauer. Im Gegensatz dazu lässt ein Accelerometer eine direkte Bestimmung der Lage zu. Dies ist möglich, da ein Accelerometer auch die Erdbeschleunigung misst, welche wiederum Rückschlüsse auf die Lage des Sensors zulässt. Da jede Bewegung des Sensors wiederum eine zusätzliche Beschleunigung darstellt, ist dies jedoch nur in Ruhe möglich. So ist eine Lagebestimmung allein mit einem Accelerometer durch anfängliche Messung der Lage in Ruhe und darauf folgende Integration der Beschleunigungsdeltas zwar denkbar, jedoch aufgrund der Trägheit und fehlender Drehratenmessung nur bedingt möglich. Die Lösung ist daher die Zusammenführung der Daten von Gyroskop und Accelerometer. Somit werden die Vorteile beider Systeme genutzt und ihre Nachteile weitestgehend gelöst. So bietet ein Gyroskop auf kurze Zeit betrachtet präzise Informationen selbst von kleinsten Winkeländerungen. Ein Accelerometer hingegen auf lange Zeit betrachtet Informationen zur absoluten Lage. Zusammenfassend ist nun also eine Methode gesucht, die in der Lage ist, Daten des Accelerometer und des Gyroskops so zu Verknüpfen, dass jederzeit auf eine absolute Lage zu schließen ist. Diese Methode wird im Allgemeinen als Sensordatenfusion bezeichnet. Zur Sensordatenfusion sind eine Vielzahl Konzepte bekannt. Im Folgenden soll jedoch nur auf den Komplementärfilter eingegangen werden. (Ohne, all-electronics.de) (Ohne, altervista.org) (Ohne, Sensordatenfusion)

4.3.5 Komplementärfilter

Der Komplementärfilter besteht im Wesentlichen aus einem Tiefpassfilter für die Accelerometerdaten und einem Hochpassfilter für die integrierten Daten des Gyroskops. Hierbei dient der Tiefpassfilter zur Reduzierung von auf den Accelerometer wirkenden Beschleunigungen und der Hochpassfilter zur Rauschunterdrückung bei den Gyroskopdaten. Dabei lassen sich Tiefpassfilter, Integration sowie Hochpassfilter auf folgende Formel vereinfachen:

$$\theta_i = (\alpha * (\theta_{i-1} + (\vartheta * \Delta t))) + \beta * \sigma \quad (8)$$

Wobei gilt:

$$\begin{aligned} \theta_i &:= \text{aktueller (gesuchter) Winkel} \\ \theta_{i-1} &:= \text{vorheriger Winkel (Integrationsteil)} \\ \alpha &:= \text{Koeffizient für den Gyroskopanteil (Hochpass Wirkung)} \\ \vartheta &:= \text{skalierte Winkeländerung gemessen am Gyroskop} \\ \Delta t &:= \text{vergangene Zeit zwischen } \theta_i \text{ und } \theta_{i-1} \text{ in Sekunden} \\ \beta &:= \text{Koeffizient für den Accelerometeranteil (Tiefpass Wirkung)} \\ \sigma &:= \text{Winkel gemessen am Accelerometer} \\ \vartheta + \sigma &= 1 \end{aligned}$$

Diese Formel lässt sich nun auf alle gesuchten Winkel anwenden. Dies sind der Roll-Winkel (Drehung um die X-Achse) und der Pitch-Winkel (Drehung um die Y-Achse). Ein weiterer Winkel ist der Yaw-Winkel (Drehung um die Z-Achse). Da die Z-Achse jedoch senkrecht zum Boden steht und somit die Erdbeschleunigung immer konstant auf sie wirkt, ist ein Messen des Yaw-Winkels durch Sensordatenfusion mit einem Accelerometer nicht möglich. Weiterhin geht aus obiger Formel neben der Bestimmung der Filterkoeffizienten, eine Aufbereitung der Sensordaten hervor. So gilt für die Bestimmung des Roll-Winkels der Accelerometer-Daten:

$$\sigma = \text{atan2}(x, z) * \frac{180}{\pi} \quad (9)$$

Und zur Bestimmung des Pitch-Winkels anhand der Accelerometer-Daten gilt:

$$\sigma = \text{atan2}\left(x, \sqrt{y^2 + z^2}\right) * \frac{180}{\pi} \quad (10)$$

Hierbei sind x, y und z die skalierten und Offset bereinigten Daten vom Accelerometer.

Die Bestimmung der Filterkoeffizienten hängt von der maximal erwarteten wirkenden Beschleunigung, dem Rauschen und dem Drift ab. Sie lassen sich in der Regel durch Abschätzen und Tests leicht finden und liegen für den Gyroskopanteil zumeist zwischen 0,95 und 0,98. Entsprechend folgen für den Accelerometeranteil Koeffizienten zwischen 0,05 und 0,02. (Ohne, all-electronics.de) (Ohne, altervista.org) (Ohne, Sensordatenfusion)

4.4 Kompass

Neben den geographischen Nord- und Südpolen besitzt die Erde auch dazu leicht verschobene geomagnetische Nord- sowie Südpole. Näherungsweise lässt sich das Erdmagnetfeld daher auch als Dipol betrachten und zur Richtungsbestimmung nutzen. Bei einem elektronischen Kompass handelt es sich somit um einen Magnetfeldsensor. Hierzu können Fluxgate-Magnetometer, Hall-Sensoren oder wie in integrierten Schaltkreisen üblich magnetoresistive Materialien genutzt werden. Bei letzteren wird der magnetoresistive Effekt, welcher eine Veränderung des elektrischen Widerstandes eines Materials bei Anlegen eines äußeren Magnetfeldes bewirkt, ausgenutzt. Zur Bestimmung des geomagnetischen Nordpols kann nun im einfachsten Fall das parallel zur Erdoberfläche verlaufene Magnetfeld in seinen horizontalen x und y Komponenten bestimmt werden. Sind diese beiden Komponenten bekannt, kann über das Verhältnis der gemessenen Feldstärken der Kurs mit folgenden Relationen bestimmt werden:

$$K(x, y) = \begin{cases} 90 \text{ für } x = 0, y < 0 \\ 270 \text{ für } x = 0, y > 0 \\ 180 - \arctan\left(\frac{x}{y}\right) * \frac{180}{\pi} \text{ für } x < 0 \\ -\arctan\left(\frac{x}{y}\right) * \frac{180}{\pi} \text{ für } x > 0, y < 0 \\ 360 - \arctan\left(\frac{x}{y}\right) * \frac{180}{\pi} \text{ für } x > 0, y > 0 \end{cases} \quad (11)$$

Aufgrund der dreidimensionalen Ausbreitung des Magnetfeldes gilt dies jedoch nur für eine Messung parallel zur Erde. Um den Kurs auch bestimmen zu können, wenn der Sensor nicht parallel zur Erde ausgerichtet ist, muss eine sogenannte Tilt-Kompensation durchgeführt werden. Hierzu muss neben der vertikalen Komponente des Erdmagnetfeldes auch die Lage des Sensors bekannt sein. Eine Tilt-Kompensation wird im Rahmen dieser Arbeit jedoch nicht umgesetzt. Auch werden Fehler aufgrund von störenden Magnetfeldern in der Umgebung nicht berücksichtigt. (Johnny Merkel, 2014) (Ohne, HMC5883L Notes)

4.4.1 Offset

Wie bei den Lagesensoren ist auch bei einem elektronischen Kompass eine Offsetbestimmung und ein Ausgleich notwendig. Der Offset setzt sich hierbei aus folgenden Komponenten zusammen:

1. Herstellungsspezifische Fehler
2. Nicht in Waage angebrachter Sensor
3. Ablenkungen des Magnetfelds durch Metall in der unmittelbaren Umgebung des Sensors

Zur Bestimmung des Offsets kann der montierte Sensor um 360° gedreht und die gemessene y Komponente in Abhängigkeit von der gemessenen x Komponente grafisch dargestellt werden. Beispielhaft ist dies in Abbildung 8 dargestellt, auf der eine leichte Verschiebung auf der x Achse zu erkennen ist. (Ohne, HMC5883L Notes)

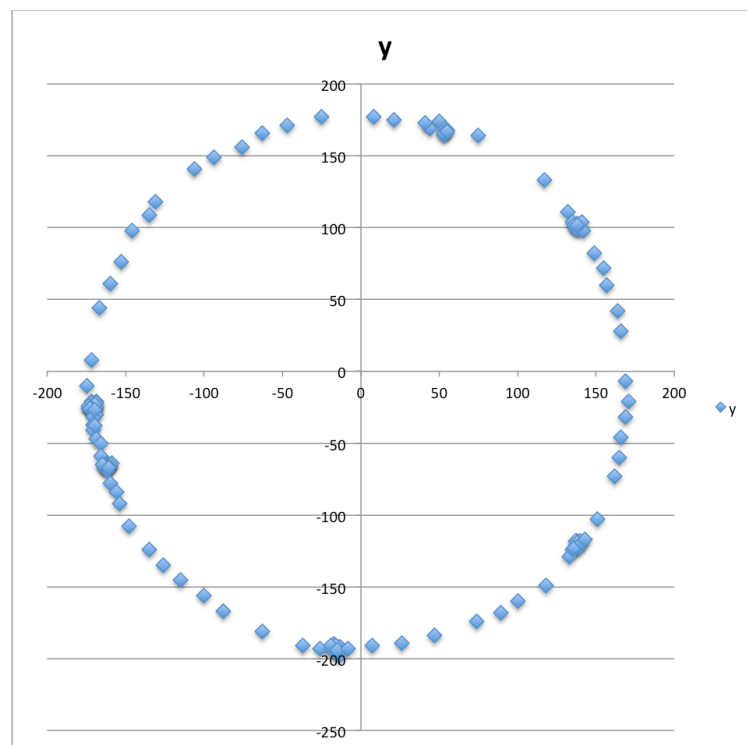


Abbildung 8 Offsetbestimmung

4.5 Pulsweiten-Modulations-Verfahren (PWM)

Mit Hilfe des PWM kann ein Leistungstransistor so geschaltet werden, dass für einen hinreichend trägen Verbraucher eine in seiner Amplitude steuerbare analoge Spannung entsteht. Hierzu wird der Transistor innerhalb einer festen Frequenz T für die Zeit t_{ein} ein- und entsprechend für die Zeit $\frac{1}{T} - t_{ein}$ ausgeschaltet (siehe Abbildung 9). Ist die Frequenz T

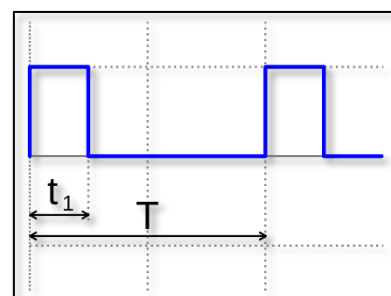


Abbildung 9 PWM (Ohne, PWM)

hinreichend groß gewählt, entsteht für kapazitive oder induktive Lasten eine Tiefpasswirkung. Diese sorgt für eine Glättung der Rechteck-Spannung und ermöglicht damit die Steuerung der Spannungsamplitude. Hierdurch lässt sich sowohl die Drehzahl regulieren als auch die maximale Motorspannung limitieren. Der Zusammenhang zwischen Spannung und PWM Signal ist in Abbildung 10 veranschaulicht. (Ohne, Pulsweitenmodulation) (Ohne, PWM)

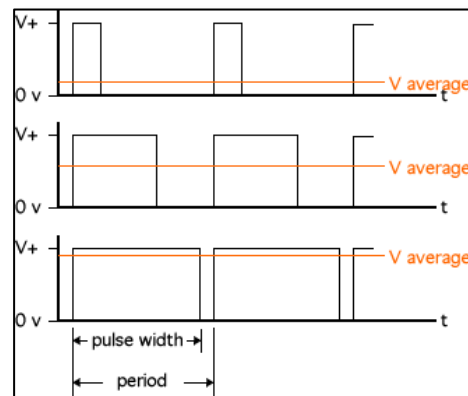


Abbildung 10 PWM Spannung (Ohne, chip.de)

Zur Limitierung der maximalen Motorspannung muss das maximale t_{ein} in Abhängigkeit von der Eingangsspannung bestimmt werden. Hierzu kann die Formel zur Bestimmung der Durchschnittsspannung eines PWM Signales umgestellt werden:

$$\bar{y} = D * y_{max} + (1 - D) * y_{min} \quad (12)$$

Hierbei geben y_{min} die Spannung für den Zustand „ausgeschaltet“ und y_{max} für den Zustand „eingeschaltet“ an. Mit D wird der sogenannte Duty-Cycle, das heißt die prozentuale Zeitspanne im Zustand „eingeschaltet“, angegeben. (Ohne, Pulsweitenmodulation) (Ohne, PWM)

5 Entwurf des Prototypen

Der zu entwerfende Prototyp soll als Grundlage für die Entwicklung eines autonomen Roboters dienen, welcher in der Lage ist, einen Raum selbstständig nach Munition zu durchsuchen. Um Kollisionen mit Objekten der Umwelt zu vermeiden und um eine gezielte Navigation um Hindernisse herum zu ermöglichen, ist eine Hinderniserkennung rund um den Roboter erforderlich. Weiterhin soll der Roboter in der Lage sein, Treppen zu erkennen, um Abstürze zu vermeiden. Zur Erkennung der Munition soll eine visuelle Objekterkennung auf Basis von opencv eingesetzt werden. Hierzu ist eine in Blickrichtung des Roboters angebrachte Kamera notwendig. Da der Roboter im Nachgang dieser Arbeit in der Lage sein soll, einen niedrigen Akkustand zu erkennen, um daraufhin gezielt eine Basisstation anzufahren, ist neben einer Akkustandsüberwachung auch eine geeignete Positionsbestimmung erforderlich. Dabei ist eine weitere Herausforderung für die Positionsbestimmung, dass diese auch in der Lage sein soll, Nutzereingriffe erkennen und kompensieren zu können. Hierfür soll ein kombiniertes Verfahren aus Odometrie und visueller Landmarkenerkennung an der Raumdecke untersucht (siehe Kapitel 6) und eingesetzt werden. Für dieses ist neben den Odometriesensoren eine weitere zur Raumdecke blickende Kamera erforderlich. Des Weiteren ist, um Nutzereingriffe zu erkennen und den Kurs auch nach diesen bestimmen zu können, ein Lagesensor sowie ein Kompass vorgesehen. Im folgendem werden die eingesetzten Sensoren sowie die gewählte Architektur des Gesamtsystems vorgestellt und erläutert.

5.1 Chassis

Da eine Anforderung ist, auch kleine Hindernisse, wie Kabel, überfahren zu können, soll im Rahmen dieser Arbeit auf ein kettengetriebenes Chassis gesetzt werden. Hierbei wurde bei einer Recherche schnell klar, dass sich das RP6 Chassis (siehe Abbildung 11) der Firma Arexx am besten eignet. Dieses bietet sowohl genug Platz für den benötigten Aufbau als auch die nötige Robustheit. Weiterhin ist das Chassis ohne die firmeneigene Elektronik sehr günstig erhältlich und bietet so maximale Freiheit bei der Gestaltung des Aufbaus und der Sensoren.

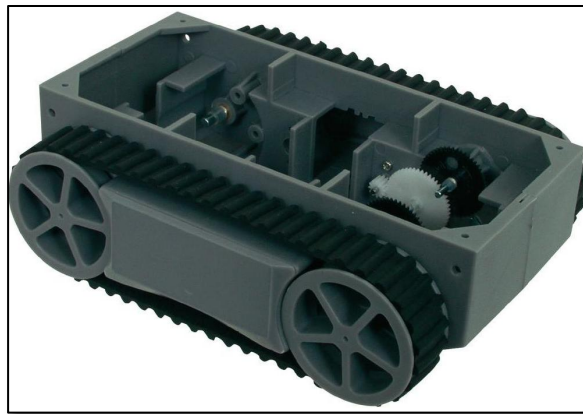


Abbildung 11 RP6 Chassis (Ohne, conrad.de)

5.2 Architektur

Die gesamte Hardware wurde weitestgehend in Modulen aufgebaut. So soll während der Entwicklung eine bessere Fehlersuche und ggf. Austausch von einzelnen Modulen ermöglicht werden. Eine Übersicht der gesamten Modularchitektur wird in Abbildung 12 gezeigt.

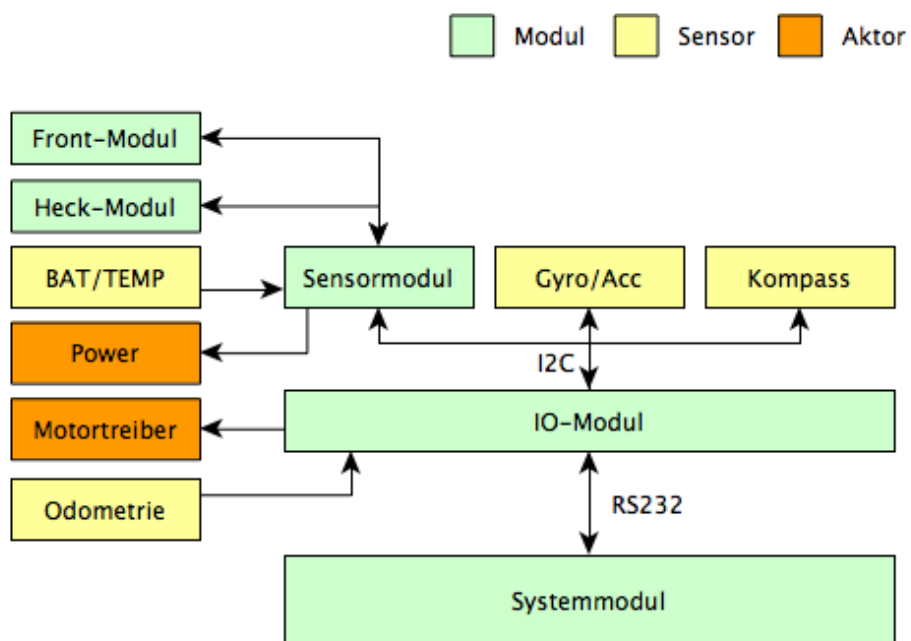


Abbildung 12 Hardware Architektur

5.3 Systemmodul

Das Systemmodul ist die Hauptrecheneinheit des Roboters und aufgrund der Anforderung eines autarken Betriebs notwendig. Auf diesem wird der Hauptteil der Software für die Positionsbestimmung, Objekterkennung und Steuerung laufen. Hierbei stellt gerade die visuelle Objekterkennung relativ hohe Anforderungen an die Prozessorleistung des Systemmodules. Weiterhin ist, um eine lange Betriebszeit des Roboters zu erreichen, eine möglichst geringe Leistungsaufnahme erforderlich. Aufgrund der Verwendung von opencv für die Objekterkennung und um auf möglichst viel vorhandene Software, wie z.B. Netzwerkstack, zurückgreifen zu können soll, das verwendete Systemmodul eine Standard Linux Distribution unterstützen. Aufgrund dieser Anforderungen sowie der erhältlichen Produktvielfalt soll ein System basierend auf einem ARM® Prozessor eingesetzt werden. Um die verfügbaren Systeme nicht weiter einzugrenzen, ist die Architektur so gewählt, dass alle Sensoren sowie die Motorsteuerung über ein per RS232 angebundenes externes IO-Modul erfolgen soll. So muss lediglich ein RS232 Port vorhanden sein und nicht auf verfügbare herausgeführte IO-Pins geachtet werden. Die Hardwareanforderungen an dieses System sind in Tabelle 1 zusammengefasst.

Anforderung	Wert
Anzahl Prozessorkerne	Min. 3
Taktfrequenz pro Kern	Min. 1000 MHz
Arbeitsspeicher	Min. 2 GB
USB Ports	Min. 2
RS232 Schnittstelle	Ja
WLAN	Ja
Größe (BxL)	Max. 13x19cm

Tabelle 1 Hardwareanforderungen

Hierbei ist die minimal erforderliche Taktfrequenz eines Kernes eine Abschätzung anhand erster Versuche mit opencv auf einem Standard PC. Die erforderlichen drei Kerne ergeben sich aus den zwei parallel laufenden Objekterkennungsprozessen, für die jeweils ein Kern vorgesehen ist, sowie einen weiteren für alle anderen Aufgaben. Die Menge an Arbeitsspeicher ist möglichst groß geschätzt, um sowohl für die Objekterkennung als auch Positionsbestimmung über ausreichend Speicher zu verfügen. Die zwei USB-Ports sind erforderlich, da als Kameras zwei Standard USB Webcams vorgesehen sind. Die maximale Größe entspricht der Größe des verwendeten Chassis. Die drei nach einer Recherche am besten passendsten Systeme sind zur Auswahl eines geeigneten Systemmodules in Tabelle 2 gegenübergestellt.

Anforderung	ODROID-U3	Wandboard-Quad	Arndale Octa
Anzahl Kerne	4	4	4+4
Taktfrequenz	1,7 GHz	1 GHz	4x1,8GHz+4x1,3GHz
Arbeitsspeicher	2 GB	2 GB	3 GB
USB Ports	3	1	3
WLAN	Nein	Ja	Extra Modul
RS232	Ja	Ja	Ja
Größe	8,3x4,8xm	9,5+9,5cm	10,4x8,5cm
Benötigte Spannung	5V	5V	5V
Max. Leistungsaufnahme	2000 mA	3000 mA	2300 mA
Preis	65,- \$	129,- \$	199,- \$

Tabelle 2 Gegenüberstellung

Es ist zu erkennen, dass das Arndale Octa die beste Prozessorleistung bietet. Jedoch wurde aufgrund des deutlich geringeren Preises und der geringeren Größe bei ausreichender Leistung das ODROID-U3 als Systemmodul ausgewählt. Da das ODROID-U3 über 3 USB-Ports verfügt, kann das fehlende WLAN Modul durch ein USB WLAN-Stick ergänzt werden. Als Betriebssystem kommt die ARM® Version der Linux Distribution Ubuntu 13.10 zum Einsatz.

5.4 IO-Modul

Das IO-Modul ist die Schnittstelle zwischen Sensoren, Motortreiber und dem Systemmodul (siehe Abbildung 13). Zur Kommunikation mit dem Gyroskop- und dem Beschleunigungssensor sowie Kompass und Sensormodul verfügt das IO-Modul

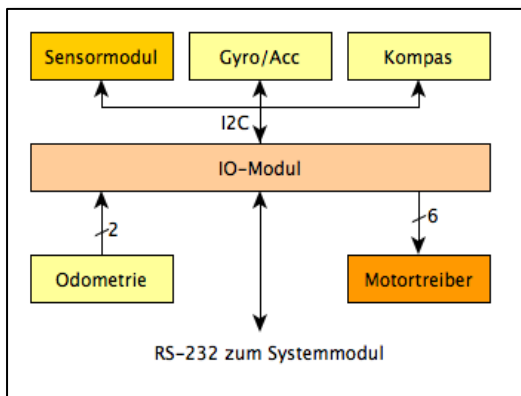


Abbildung 13 IO-Modul Anbindung

über einen i2c Bus. Die Anbindung der Odometriesensoren erfolgt über zwei TTL IO-Ports mit Flankenwechsel Interrupt. Zur Steuerung der Motoren sind des Weiteren vier einfache TTL sowie zwei mit PWM Funktion ausgestatte IO-Ports vorgesehen. Eine direkte Anbindung der Sensoren und der Motosteuerung wäre auch über den vorhanden i2c Bus des Systemmodules möglich. Da dieser mit einer Spannung von 1,8V arbeitet, die Sensoren jedoch mit 5V, wäre hier eine Transferschaltung notwendig. Des Weiteren ist für die Odometriesensoren ein Zähler notwendig, welcher auf jeden Flankenwechsel schnellstmöglich reagiert. Dies ist mit einem dedizierten Mikrocontroller auf dem IO-Modul relativ einfach zu implementieren. Auf dem Systemmodul würde dies

hingegen den Einsatz eines Echtzeitbetriebssystems erfordern. Daher wird als IO-Modul ein für das Systemmodul erhältliches IO-Expansion-Modul eingesetzt. Dieses verfügt über einen Atmel Mega 328p Mikrocontroller, welcher alle Anforderungen erfüllt und platzsparend auf das Systemmodul aufgesteckt werden kann.

5.5 Sensoren

5.5.1 Hinderniserkennung

Aufgrund der hohen Reichweite, des breiten Öffnungswinkels und der geringen Wahrscheinlichkeit auf absorbierende Materialien zu treffen, wird im Rahmen dieser Arbeit zur Hinderniserkennung und Kollisionsvermeidung auf Ultraschallsensoren gesetzt. Hierzu dienen jeweils zwei SRF05 Module für die Front und das Heck sowie jeweils zwei SRF04 Module für die linke und rechte Seite. Die Wahl der Module ist einzig auf den Preis reduziert. Hierbei wird davon ausgegangen, dass für die Front und das Heck eine höhere Reichweite als an den Seiten benötigt wird. Daher werden an den Seiten die günstigeren SRF04 Module eingesetzt. Die Module weisen die in Tabelle 3 zusammengefassten Leistungsdaten auf.

Bezeichnung	SRF05	SRF04
Spannungsversorgung	5V	5V
Typische Leistungsaufnahme	4mA	50mA
Max. Reichweite	4m	3m
Min. Reichweite	1cm	3cm
Frequenz	40kHz	40kHz
Interface	TTL	TTL
Größe	43x20x17mm	43x20x17mm

Tabelle 3 Leistungsdaten

Das Interface zur Ansteuerung der Module ist eine einfache TTL Schaltung. Hierzu verfügt jedes Modul über einen Trigger- und einen Echo Pin. Wie in Abbildung 14 gezeigt muss zur Auslösung einer Messung der Trigger Pin für mindestens 10 μ s auf High geschaltet werden. Das Modul setzt danach das modulierte Ultraschall-Signal ab und startet eine Zeitmessung. Erkennt das Modul das Wiedereintreffen des ausgesendeten Signals, schaltet es den Echo Pin für die gemessene Zeitspanne auf High. Das heißt zur Bestimmung der Entfernung eines erkannten Objektes kann die Zeitspanne, die der Echo Pin auf High geschaltet ist, gemessen und das Ergebnis in die Formel 6 eingesetzt werden. Da der Schall in Luft bei 20°C ca. 8m in 23ms zurücklegt, was einem Objekt in einer Entfernung von ca. 4m entsprechen würde, gilt diese Zeitspanne für die SRF05 Sensoren als „kein Objekt gefunden“. Für die

SRF04 Module gilt analog dazu eine Zeitspanne von mehr als 17ms als „kein Objekt gefunden“. Um Fehlmessungen aufgrund von Echos vorangegangener Messungen zu vermeiden, dürfen die Module maximal alle 50ms eine Messung starten. Eine gleichzeitige Messung aller Module ist ebenfalls aufgrund einer möglichen Fehlmessung durch gegenseitige Beeinflussung nicht vorgesehen. Da die einzelnen Module somit nur sequentiell genutzt werden können, ergibt sich ein Messintervall von ca. zwei Messungen pro Sekunde. (Ohne, SRF04 Specs) (Ohne, SRF05 Specs)

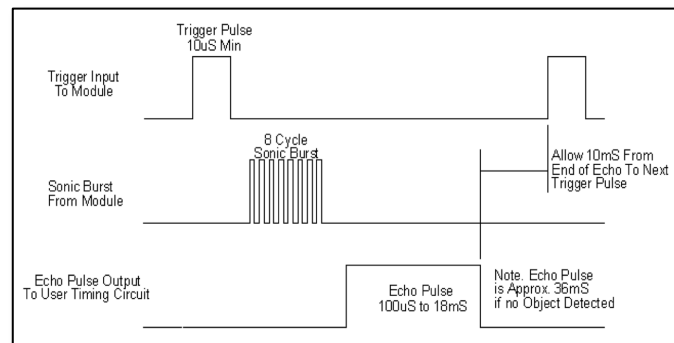


Abbildung 14 TTL-Interface (Ohne, SRF05 Specs)

5.5.2 Treppendetektion

Aufgrund des relativ schmalen Abstrahlwinkels und des höheren Messintervalls sind Infrarotsensoren für die Erkennung von Senken besser geeignet als Ultraschallsensoren. So können Infrarotsensoren näher am Roboter montiert werden, ohne von Reflexionen des Chassis beeinflusst zu werden, und sind somit in der Lage, Senken früher zu erkennen. Des Weiteren ist der größere Messintervall für eine schnelle Erkennung von Senken und Reaktion darauf von Vorteil. Daher werden im Rahmen dieser Arbeit für die Erkennung von Senken Infrarotsensoren eingesetzt. Da sich nach einer kurzen Recherche herausstellte, dass Infrarot-Distanzsensoren der Firma Sharp® die besten Eigenschaften bei Genauigkeit und Toleranz gegenüber Fremdlicht sowie Reflexionsoberfläche bieten, werden hier nur solche eingesetzt. Das einzige Auswahlkriterium ist daher der Messbereich des Sensors. Da es sich bei dem Roboter um eine kettengetriebene Plattform handelt, können Senken bzw. Erhöhungen von maximal 2cm Tiefe bzw. Höhe problemlos überwunden werden. Dies entspricht der Hälfte der Kettenhöhe abzüglich 0,5cm Sicherheitspuffer. Die Infrarotsensoren sollen in einem Abstand zum Boden von ca. 8cm angebracht werden. Daher muss der Messbereich der Infrarotsensoren mindestens zwischen 6 und 10cm liegen. Das Sharp® Infrarotsensor Model GP2D120 ist mit einem Messbereich von 4-30cm somit am besten geeignet und wird daher für die Treppendetektion eingesetzt. Zur zuverlässigen Erkennung von potentiell gefährlichen Senken werden zwei Sensoren jeweils direkt vor jeder Kette

an der Front sowie am Heck angebracht. Da diese Sensoren eine analoge Spannung in Abhängigkeit zur gemessenen Entfernung ausgeben, sind weiterhin vier Analog-Digital-Konverter erforderlich (ADC). Eine Zusammenfassung der Leistungsdaten des GP2D120 Infrarotsensors ist in Tabelle 4 geliefert. (Ohne, roboticlab.eu) (Ohne, GP2D120 Specs)

Merkmal	Wert
Spannungsversorgung	5V
Max. Leistungsaufnahme	50mA
Messbereich	4-30cm
Max. Messfrequenz	Ca. 25Hz
Ausgang	0,4-2,6V

Tabelle 4 Leistungsdaten

Wie in Abbildung 15 zu erkennen ist der Zusammenhang zwischen ausgegebener Spannung und Entfernung nicht linear. Auch liefert Sharp® im Datenblatt des Sensors keine Funktion zur Berechnung der Entfernung anhand der ausgegebenen Spannung. Daher wird eine solche Funktion näherungsweise bestimmt. Hierzu wird eine Polynomfunktion anhand von Referenzwerten aufgestellt. Diese wird zum Ausgleich von Herstellungsbedingten Unterschieden für jeden Sensor getrennt bestimmt. Hierzu wird die „Curve-Fitting App“ von Matlab® eingesetzt und eine Polynomfunktion 4. Grades bestimmt. Das Curve-Fitting Verfahren selbst soll hier keine Rolle spielen.

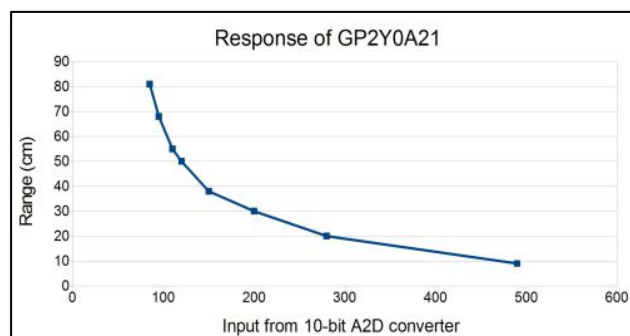


Abbildung 15 Spannung/Distanz (Ohne, acroname.com)

Um eine möglichst hohe Genauigkeit im relevanten Messbereich zwischen 6 und 10cm Entfernung zu erreichen, wird dieser Bereich bei der Referenzpunktmessung bevorzugt. In Tabelle 5 ist exemplarisch eine Messung dargestellt. (Ohne, acroname.com)

Entfernung (mm)	Spannung (V)
40	2,72
60	2,01
70	1,75
80	1,58
90	1,4
100	1,28
120	1,06
160	0,81
220	0,6
300	0,43

Tabelle 5 Referenzmessung

Anhand der Referenzpunkte aus Tabelle 5 lässt sich nun folgende Polynomfunktion als Näherung von $R(V)$ (Range über Volt) bestimmen:

$$R(V) = -28.81x^5 + 260.2x^4 - 929.9x^3 + 1672x^2 - 1584x + 738.5 \quad (13)$$

5.5.3 Temperatursensor

Da die Schallgeschwindigkeit temperaturabhängig ist, wird ein Temperatursensor zur genaueren Entfernungsmessung mit Hilfe der Ultraschallsensoren eingesetzt. Hierzu kommt ein temperaturabhängiger Widerstand vom Typ KTY 81-220 zum Einsatz. Dieser ist mit einem Festwiderstand von 2700 Ohm in Reihe geschaltet und mit der 5V Versorgungsspannung verbunden. Über die an dem Temperatursensor abfallende Spannung kann dessen Widerstand bestimmt und somit die Temperatur berechnet werden. Der Temperatursensor ist hierfür an einem ADC-Port des IO-Moduls angeschlossen. Zur Bestimmung des Widerstandes dient folgende nach dem ohmschen Gesetz umgestellte Formel: (Ohne, sprut.de)

$$R = 2700 * \frac{U_{gemessen}}{U_{gesamt} - U_{gemessen}} \quad (14)$$

Zur Berechnung der Temperatur in Abhängigkeit vom Widerstand wurde mit Hilfe der „Curve-Fitting App“ von Matlab® und den Referenzwerten aus dem Datenblatt folgende Polynomfunktion als Näherung bestimmt: (Ohne, KTY-81 specs)

$$T = -175.32 + (0.151 * R) - (0.00003202 * R^2) + (0.000000003355 * R^3) \quad (15)$$

5.5.4 Lagesensor

Um einen Nutzereingriff zu erkennen, soll eine dreidimensionale Lageerkennung eingesetzt werden. So soll bei einer nicht erwarteten Lageänderung, beispielweise während der Roboter still steht, ein Nutzereingriff erkannt werden. Auch sind so während der Fahrt des Roboters Eingriffe anhand nicht zur eigenen Bewegung passender Lageänderungen erkennbar. Hierfür müssen im Nachgang Grenzwerte für die Bewegung in Abhängigkeit von Zeit, Zustand (stehend, fahrend) und Achse experimentell bestimmt werden. Zur Lageerkennung kommt ein Modul mit dem integrierten drei achsigen Gyroskop und dem drei achsigen Accelerometer Sensor MPU-6050 der Firma InvenSense zum Einsatz. Dieses ist sehr günstig zu erhalten und aufgrund des vorhandenen i2c Buses und 5V Spannungsversorgung einfach in die Hardware zu integrieren. Die Konfiguration und Auswertung erfolgt über Registerzugriffe durch das IO-Modul. Da für die Integration der beiden Sensoren mittels Komplementärfilter möglichst genaue Zeit-Deltas erforderlich sind, erfolgt diese direkt auf dem IO-Modul. (Ohne, MPU6050 specs)

5.5.5 Kompass

Aufgrund der Anforderung, auch nach einem Nutzereingriff eine Positionsbestimmung durch- bzw. weiterführen zu können, ist eine Kursbestimmung mittels Odometrie nicht möglich. Daher wird zur Bestimmung des Kurses ein elektronischer Kompass eingesetzt. So sind auch nach einem Nutzereingriff die Blickrichtung und damit auch der Kurs des Roboters erkennbar. Als Kompass kommt hierzu ein Modul mit dem drei achsigen Sensor HMC5883L der Firma Honeywell zum Einsatz. Wie der Lagesensor ist dieses Modul sehr günstig zu beschaffen und kann aufgrund seines i2c Bus und einer 5V Spannungsversorgung einfach in die Hardware integriert werden. Auch bei dem Kompassmodul erfolgen Konfiguration und Auswertung durch Registerzugriffe durch das IO-Modul. Zur Berechnung des Kurses wird im Rahmen dieser Arbeit keine Tilt-Kompensation durchgeführt. Dies hat lediglich zur Folge, dass Tests der Sensoren nur in Waage durchgeführt werden können. (Ohne, HMC5883L Notes)

5.5.6 Odometrie

Als Odometriesensoren werden zwei Inkrementaldrehgeber auf Basis einer Gabellichtschranke eingesetzt. Diese sind als Ersatzteil passend für das verwendete Chassis erhältlich und erzeugen 626 Flankenwechsel pro Radumdrehung. Aufgrund des Radumfangs und der Auflösung ergibt sich für jeden Flankenwechsel eine zurückgelegte Strecke von ca. 0,23mm. (Ohne, RP6 Encoder)

5.5.7 Sensormodul

Beim Sensormodul handelt es sich um eine selbst entwickelte Platine, welche im wesentlichen aus einem Atmel® Mega 8 Mikrocontroller sowie dem Lagesensor und Kompass besteht. Auf dem verbauten Mikrocontroller werden die Temperatur-, Ladestands-, Infrarot- und Ultraschallmessungen durchgeführt und können durch das IO-Modul per i2c Bus abgerufen werden. Des Weiteren ist das IO-Modul in der Lage den gesamten Roboter abzuschalten, sollte der Akku einen kritischen Ladestand erreichen. Hierzu ist der Spannungsregler, der alle Komponenten mit Strom versorgt, über einen Leistungstransistor mit dem Akku verbunden. Das IO-Modul hält diesen Transistor über einen IO-Port im geschalteten Zustand und ermöglicht so die Stromversorgung. Somit ist die gesamte Stromversorgung über einen Totmannschalter realisiert und kann vom Sensormodul jederzeit unterbrochen werden. Zur Kommunikation mit dem IO-Modul ist das Sensormodul über einen i2c Bus mit diesem verbunden. Außerdem wurden der Lagesensor und der Kompass auf diesem Modul verlötet und werden von diesem mit Strom versorgt sowie mit dem i2c Bus verbunden. Der Schaltplan des Moduls kann dem Anhang dieser Arbeit entnommen werden.

5.5.8 Heckmodul

Wie das Sensormodul ist auch das Heckmodul eine Eigenentwicklung. Auf diesem Modul sind zwei SRF05 und zwei SRF04 Ultraschallsensoren zur Hinderniserkennung sowie zwei nach unten gerichtete Sharp® GP2D120 Infrarot Distanzsensoren zur Treppendetektion untergebracht (siehe Abbildung 16). Zur einfachen Verbindung des Heckmodules mit dem Sensormodul verfügt das Heckmodul des Weiteren über einen Atmel® Tiny2313 Mikrocontroller. Dieser stellt einen in Software umgesetzten Multiplexer sowie Demultiplexer für die Ultraschallsensoren bereit. So kann über zwei Select Ports ein Sensor ausgewählt und nach Schalten des Enable Ports der Trigger des gewählten Sensors für 10µs ausgelöst werden. Das Echo Signal des ausgewählten Sensors wird über einen Echo Port zurück an das Sensormodul durchgereicht. So konnte die Anzahl nötiger Verbindungen für die

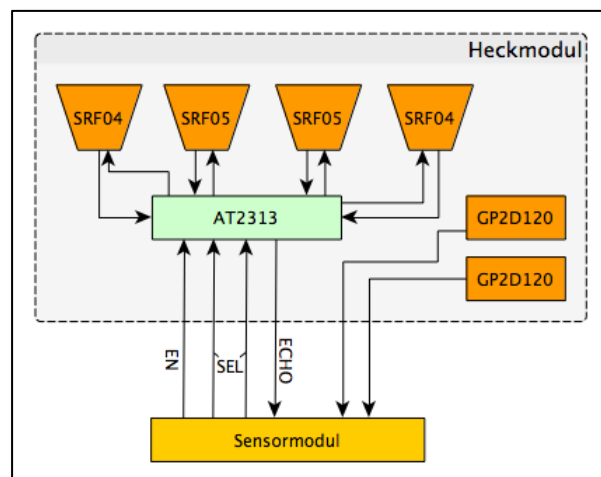


Abbildung 16 Heckmodul Anbindung

Ultraschallsensoren auf vier halbiert werden. Die beiden Infrarot Distanzsensoren sind über zwei weitere Leitungen mit dem Sensormodul verbunden. So kann das gesamte Heckmodul über ein sechs poliges Flachbandkabel platzsparend mit dem Sensormodul verbunden werden. Zur weiteren Reduzierung der benötigten Interrupt-fähigen Ports auf dem Sensormodul ist der Echo Port des Heckmodules als Tristate implementiert. So wird der Echo Port in einen hochohmigen Zustand geschaltet, sobald der Enable Port nicht geschaltet ist. Daher können die Echo Ausgänge des Heck- und des Frontmodules auf Seiten des Sensormodules auf den gleichen Port geschaltet werden. Der Schaltplan des Moduls kann dem Anhang dieser Arbeit entnommen werden.

5.5.9 Frontmodul

Auch das Frontmodul ist eine Eigenentwicklung und enthält neben den Komponenten des Heckmodules die Schaltung für die gesamte Stromversorgung des Roboters. Dies ist, neben einem Spannungsregler für die Erzeugung der 5V Betriebsspannung, ein vorgeschalteter Leistungstransistor sowie ein Taster zur manuellen Überbrückung des Transistors (Einschalten des Roboters). Weiterhin sind auf diesem Modul der Temperatursensor und ein Spannungsteiler zur Messung der Akkuspannung untergebracht. Diese werden direkt mit jeweils einem ADC Port des Sensormodules verbunden. Der Temperatursensor dient hierbei der zuvor erwähnten Temperatorkompensation der Ultraschallsensoren und der Spannungsteiler zur Bestimmung des Akku-Ladestandes. Ein Spannungsteiler ist nötig, da der Akku eine Spannung zwischen 8,4V und 6,4V liefert, alle verfügbaren ADC Ports jedoch eine maximale Spannungsmessung bis 5V erlauben. Der Schaltplan des Moduls kann dem Anhang dieser Arbeit entnommen werden.

5.6 Antrieb

Beim Antrieb handelt es sich um zwei gewöhnliche Gleichstrommotoren mit einer Betriebsspannung von jeweils 6V. Diese sind jeweils über ein Getriebe mit einer Kette verbunden und ermöglichen somit die getrennte Ansteuerung jeder Kette. Als Motor-Treiber IC wird der L293D verwendet. Hierbei handelt es sich um zwei Brückenschaltungen mit jeweils zusätzlicher Enable-Schaltung und Freilaufdioden. Eine Brückenschaltung (siehe Abbildung 17) besteht hierbei im Wesentlichen aus vier Leistungstransistoren. Diese

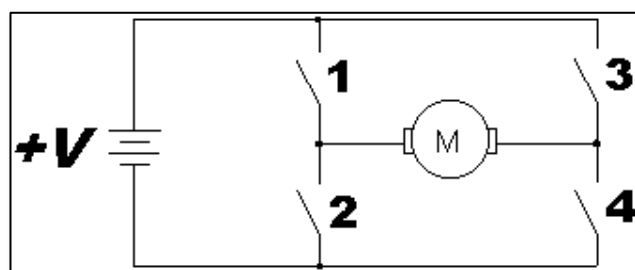


Abbildung 17 Brückenschaltung (Ohne, amateurfunkbasteln.de)

erlauben zum einen die Ansteuerung des Motors mit einer von der Logikschaltung unabhängigen Spannungsquelle als auch das Umschalten der Drehrichtung des Motors. Die Freilaufdioden dienen zum Schutz der Schaltung und gewährleisten so das Abfließen von am Motor induzierten Strömen und Spannungsspitzen. Mit Hilfe der Enable-Schaltung lässt sich die Brücke ohne Kurzschluss ein- bzw. ausschalten. Hierdurch wird eine PWM Ansteuerung und damit Drehzahlregulierung des Motors möglich. Da im L293D zwei vollständige Brückenschaltungen integriert sind, lassen sich beide Motoren mit diesem einen IC betreiben. Die maximale Treiberspannung und maximaler Stromfluss pro Brücke des L293D sind mit 36 V bzw. 600 mA angegeben. Dies ist ausreichend, da die verwendeten Motoren eine maximale Betriebsspannung von 6V und Leistungsaufnahmen unter Last von ca. 350mA aufweisen. Die Ansteuerung des Motortreibers erfolgt im IO-Modul über jeweils ein Enable und zwei Schaltsignale pro Motor. Der Motortreiber wurde zusammen auf einer kleinen Platine untergebracht, welche über ein zehn poliges Flachbandkabel mit dem IO-Modul und über einen zwei poligen Stecker mit dem Akku verbunden ist. Das Modul wird dabei auch als Verbindung zu den Odometriesensoren genutzt, welche über das Modul angeschlossen und mit Strom versorgt werden. So wird über das zehn polige Flachbandkabel sowohl die 5V Spannung geliefert als auch die Odometriesensoren und der Motortreiber mit dem IO-Modul verbunden. Der Schaltplan der Motortreiberplatine kann dem Anhang dieser Arbeit entnommen werden. (Ohne, amateurfunkbasteln.de) (Ohne, RP5/6 Chassis)

5.7 Spannungsversorgung

Als Spannungsquelle für den Betrieb dient ein handelsüblicher 7,4V Lithium Polymer (LiPo) Modellbau-Akku mit einer Kapazität von 3800mAh. Die Wahl auf diese Leistungsdaten ist in den verwendeten Komponenten begründet. So wird für das Systemmodul, die Sensorik und Logikschaltung des Motortreibers eine Spannung von 5V benötigt, für die Motoren selbst jedoch 6V. Die 5V Spannung lässt sich hierbei beispielsweise mittels eines geeigneten Festspannungsreglers aus den 7,4V gewinnen. Zur Einhaltung der 6V Motorspannung kann eine PWM Steuerung genutzt werden.

5.7.1 LiPo Akku

Bei dem verwendeten LiPo Akku handelt es sich um ein herkömmliches Modellbau Akku-Pack bestehend aus zwei in Reihe geschalteten LiPo Zellen. Jede dieser Zellen verfügt über eine Kapazität von 3800mAh und einer Nennspannung von 3,7V. Hieraus ergibt sich als gesamte Nennspannung des Akku-Packs ca. 7,4V und eine Kapazität von 3800mAh. Wie bei allen Akkus wird die Nennspannung jedoch nicht konstant über den gesamten Entladevorgang geliefert. Vielmehr ist die Spannung

des Akkus abhängig von der Last und dem Ladezustand des Akkus (siehe Abbildung 18). Für LiPo Akkus gelten als Ladeschlussspannung ohne Last ca. 4,2V pro Zelle und als Entladeschlussspannung 3,0V pro Zelle. Dabei entspricht 4,2V dem Zustand „Voll geladen“ und 3,0V als „Vollständig entladen“. Die Lade- bzw. Entladeschlussspannung des gesamten Akku-Packs entspricht demnach dem Produkt aus der Anzahl Zellen und den Spannungen pro Zelle. Die meisten Hersteller empfehlen jedoch, ihre LiPo Akkus nicht unter 20% der Kapazität zu entladen. Daher sollte eine Entladeschlussspannung von ca. 3,2V angenommen werden. Aufgrund des Zusammenhanges zwischen Spannung und Kapazität des Akkus kann der Ladezustand genähert mit folgender Formel bestimmt werden (Ohne, elektromodellflug.de) (Ohne, dexgo.com):

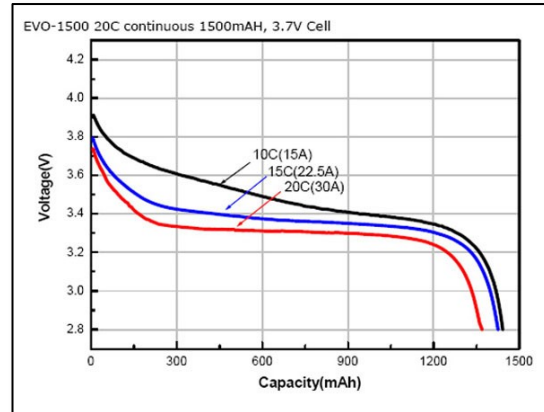


Abbildung 18 Entladekurve LiPo Akku (Ohne, re-network.de)

$$l = \frac{100}{U_{voll} - U_{leer}} * (U_{gemessen} - U_{leer}) \quad (16)$$

Hierbei zeigt l die Restkapazität in % an, U_{voll} entspricht der Ladeschlussspannung des gesamten Akku-Packs und U_{leer} der Entladeschlussspannung. Ein Problem dieser Restkapazitätsbestimmung ist jedoch, dass hierbei von einer linearen Entladekurve ausgegangen wird. Die Entladekurve eines LiPo Akkus ist jedoch typischerweise S-Kurven artig. Die sich daraus ergebene Komplexität bei der Bestimmung der Restlaufzeit soll im Rahmen dieser Arbeit durch die Wahl einer größeren Entladeschlussspannung umgangen werden. Hierzu wird eine Entladeschlussspannung von ca. 3,3V als ausreichender Puffer angesehen. Dies entspricht einer Entladung des Akkus auf ca. 25% der Kapazität. (Ohne, elektromodellflug.de)

5.7.2 5V Versorgungsspannung

Zur richtigen Dimensionierung des Spannungsreglers für die 5V Versorgungsspannung sollen die in Tabelle 6 gezeigten Verbrauchsabschätzungen dienen. Da es nicht zu allen Komponenten Datenblätter mit entsprechender Angabe der Leistungsaufnahme gibt, handelt es sich hierbei um Schätzungen anhand der Datenblätter der verwendeten Teile, eigene Erfahrungswerte und großen Puffern.

Komponente	Max. Stromverbrauch (mA)
Systemboard inkl. USB Kameras	2000
IO Modul	60
Sensor-Modul	60
Heck-Modul	245
Front-Modul	250
Motortreiber	400
Gesamt	3015

Tabelle 6 Abschätzung der Leistungsaufnahme

Wie in Tabelle 6 zu sehen liegt die geschätzte maximale Leistungsaufnahme bei ca. 3000mA. Aufgrund dieser Daten und seiner einfachen Handhabung wurde der Festspannungsregler LM1084-it5.0 ausgewählt. Dieser bietet bei einem hohen Eingangsspannungsbereich von 6,5V bis hin zu 30V und einer maximal gelieferten Leistung von 5000mA die benötigte stabile Ausgangsspannung von 5V. Die minimal benötigte Eingangsspannung von 6,5V stellt hierbei kein Problem dar, da der verwendete Akku bei einer Restspannung von 6,4V als „Vollständig entladen“ gilt. (Ohne, SRF04 Specs) (Ohne, SRF05 Specs) (Ohne, RP5/6 Chassis) (Ohne, RP6 Encoder) (Ohne, GP2D120 Specs) (Ohne, HMC5883L Notes) (Ohne, MPU6050 specs) (Ohne, t2313 specs) (Ohne, m8 specs) (Ohne, LM1084 specs)

5.8 Kameras

Sowohl für die Objekterkennung zur Suche der Munition als auch für die Positionsbestimmung an der Raumdecke kommen Standard USB-Webcams zum Einsatz. Hierbei handelt es sich in beiden Fällen um Creative® Live! Cam Sync HD mit einer Auflösung von 1280x720 Punkten. Ausschlaggebend für die Wahl dieser Modelle sind die Unterstützung des eingesetzten Betriebssystems, eine Autofocus-Funktion sowie der für die gelieferte Auflösung günstige Preis. Der Autofocus ist gerade für die Positionsbestimmung an der Raumdecke hilfreich, da so auch bei unterschiedlich hohen Räumen keine manuelle Fokussierung durch den Benutzer erforderlich ist. (Ohne, Cam HD specs)

5.9 Software

Zur Durchführung von Systemtests und zur Untersuchung des Positionierungsverfahrens wurde die Software zur Ansteuerung der Sensoren und des Antriebs entwickelt und wird im Folgenden kurz erläutert. Die gesamte Software ist auf der beigefügten CD im Verzeichnis „source“ zu finden.

5.9.1 Front-/Heckmodul

Für das Front- und das Heckmodul wurde ein in Software umgesetzter Multiplexer für die Ultraschallsensoren entwickelt. Hier läuft nach Konfiguration der Ein- und Ausgangsports des Modules und der Sensoren das in Abbildung 19 dargestellte Programm in einer Endlosschleife ab. Die Software ist auf der beigefügten CD im Verzeichnis „source/sens_back“ zu finden.

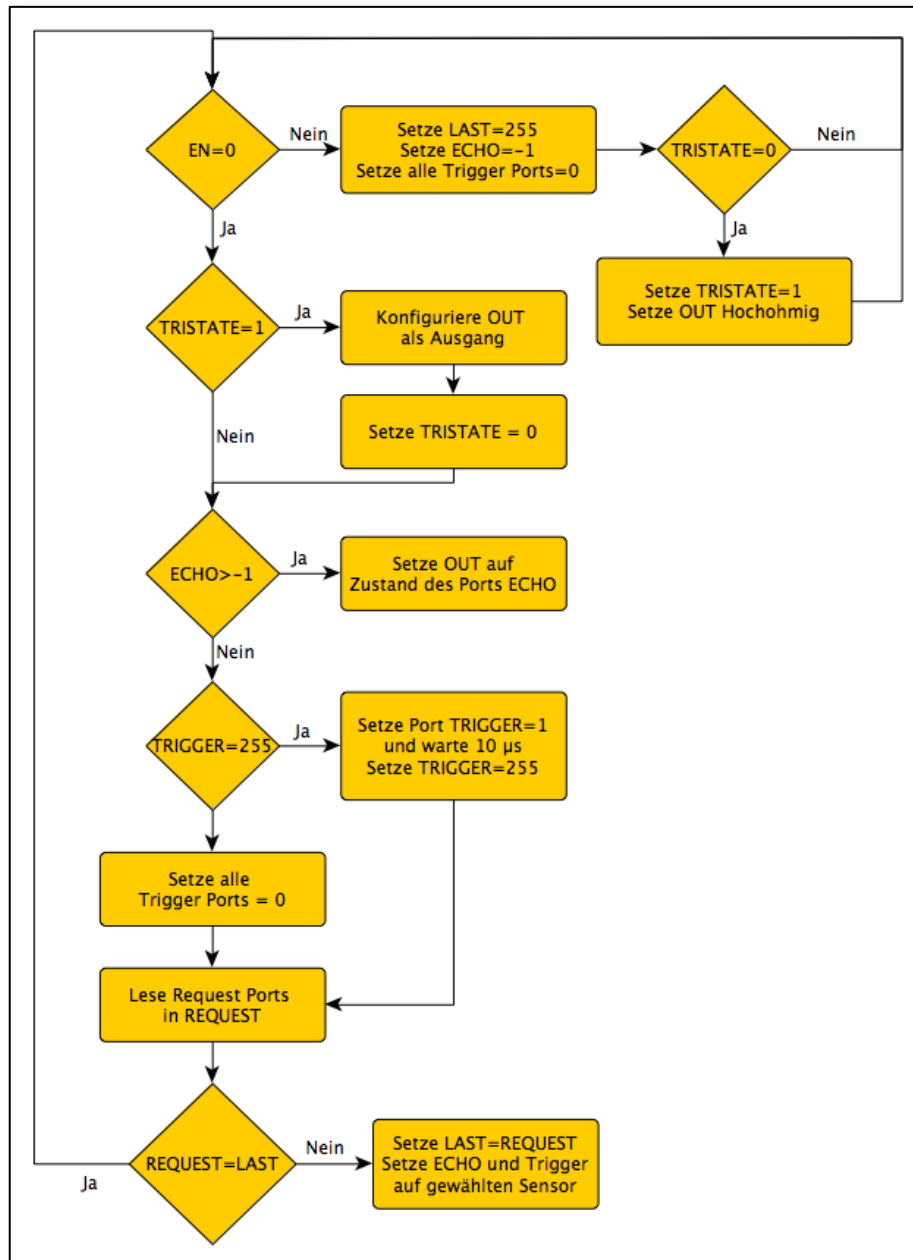


Abbildung 19 Front-/Heckmodul Ablauf

5.9.2 Senormodul

Das Sensormodul ist mit dem Front- und dem Heckmodul verbunden und dient primär der Abstraktion der Sensoren. Hier werden auf dem verbauten Mikrocontroller die Ultraschallsensoren sequentiell angesteuert und die entsprechenden Laufzeiten gemessen. Außerdem werden an den verfügbaren ADC-Ports die Spannungen der Infrarotsensoren, des Temperatursensors und des Spannungsteilers für die Akkustandskontrolle gemessen. Die gesammelten Sensordaten können als zusammengefasstes Datenpaket per i2c Bus vom IO-Modul abgerufen werden. Das Sensormodul stellt daher einen i2c Slave dar. Die für das Sensormodul entwickelte Software ist auf der beigegeführten CD unter „source/sens“ zu finden. Die Software ist in drei unabhängig voneinander laufenden Prozessen für die Ultraschallsensoren, die Spannungsmessungen sowie dem Hauptprogramm unterteilt. Diese werden im Folgenden erläutert.

Ultraschallsensoren

Die Ansteuerung und Auswertung der Ultraschallsensoren erfolgt über einen zeitgesteuerten und einen Flankenwechselinterrupt für den Eingangspin, an dem die Echo-Ausgänge des Front- und des Heckmoduls geschaltet sind. Für den zeitgesteuerten Interrupt wird ein Timer so konfiguriert, dass dieser alle 50ms eine Interrupt-Service-Routine ausführt. So wird die im Datenblatt der Ultraschallsensoren angegebene Zeit eingehalten, um den Empfang von Echos vorangegangener Messungen zu vermeiden. Bei der Ausführung der Routine wird das Ergebnis des zuvor angesteuerten Sensors gespeichert und der Prozess mit dem nächsten Sensor fortgesetzt. Sind alle acht Sensoren ausgewertet, werden die Ergebnisse in einen global verfügbaren Speicherbereich übertragen und der Ablauf erneut begonnen. Zur Zeitmessung dient der zuvor konfigurierte Timer. So wird bei einem Flankenwechsel des Echo-Eingangs der Zähler des Timers auf 0 gesetzt, falls der Echo-Eingang auf 1 umgeschaltet hat. Schaltet der Echo-Eingang auf 0, wird der Zählerstand des Timers als Ergebnis gespeichert. Da die Taktfrequenz des Timers bekannt ist, kann der Zählerstand anschließend in eine Zeitspanne umgerechnet werden. Der Ablauf ist in Abbildung 20 schematisch dargestellt.

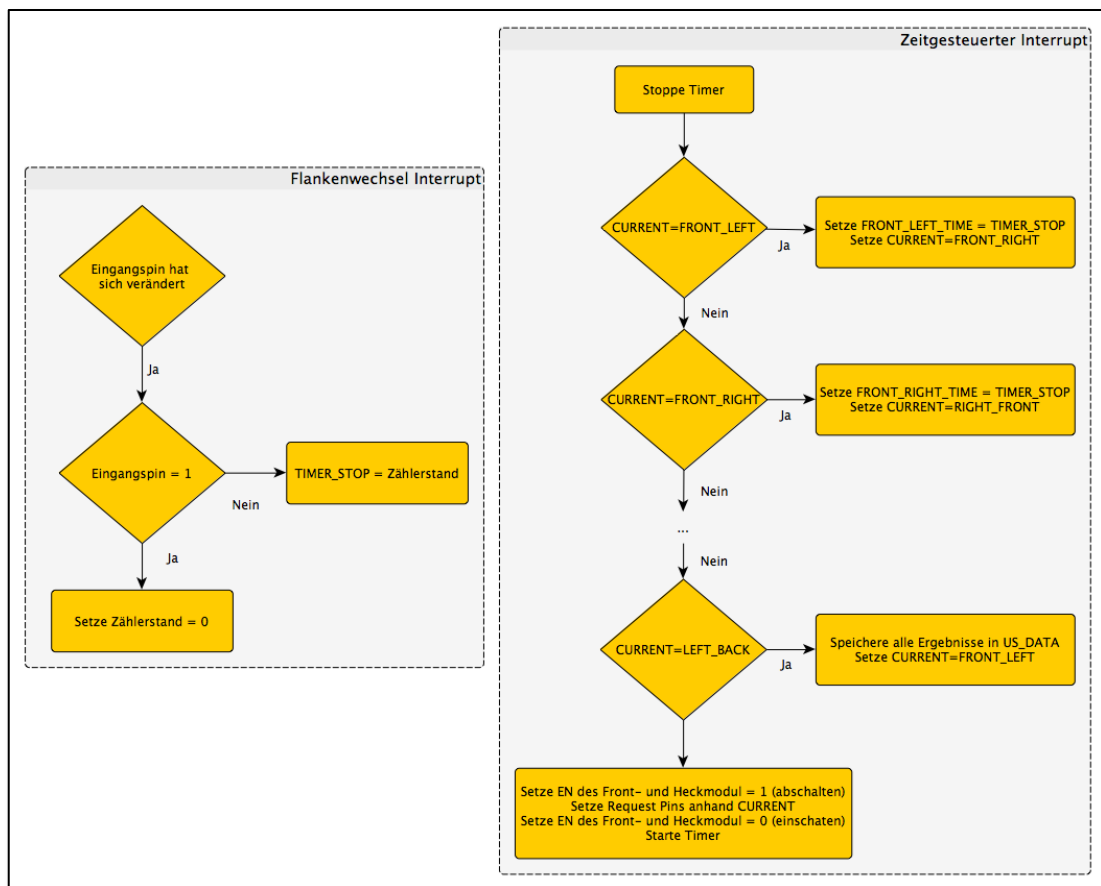


Abbildung 20 Ablauf Ultraschallsensoren

Spannungsmessung

Da die Infrarotsensoren eine zur Entfernung proportional abhängige Spannung ausgeben, werden diese über Analog-Digital-Wandler des verbauten Mikrokontrollers ausgewertet. Neben den Infrarotsensoren werden so auch die Spannungen des Temperatursensors und des Akkus gemessen. Der Ablauf ist hier ähnlich der zeitgesteuerten Komponente bei den Ultraschallsensoren. Jedoch wird der Interrupt hier nicht durch einen Timer ausgelöst, sondern direkt durch den Analog-Digital-Wandler, sobald dieser mit einer Messung fertig ist. Die Ergebnisse der Messungen werden auch hier in einen global verfügbaren Speicherbereich übertragen. Des Weiteren wird nach der Messung des Akkustandes dieser auf einen Schwellwert überprüft und ggf. die Spannungsversorgung unterbrochen. Der Ablauf bei auftreten eines ADC Interrupts ist in Abbildung 21 schematisch dargestellt.

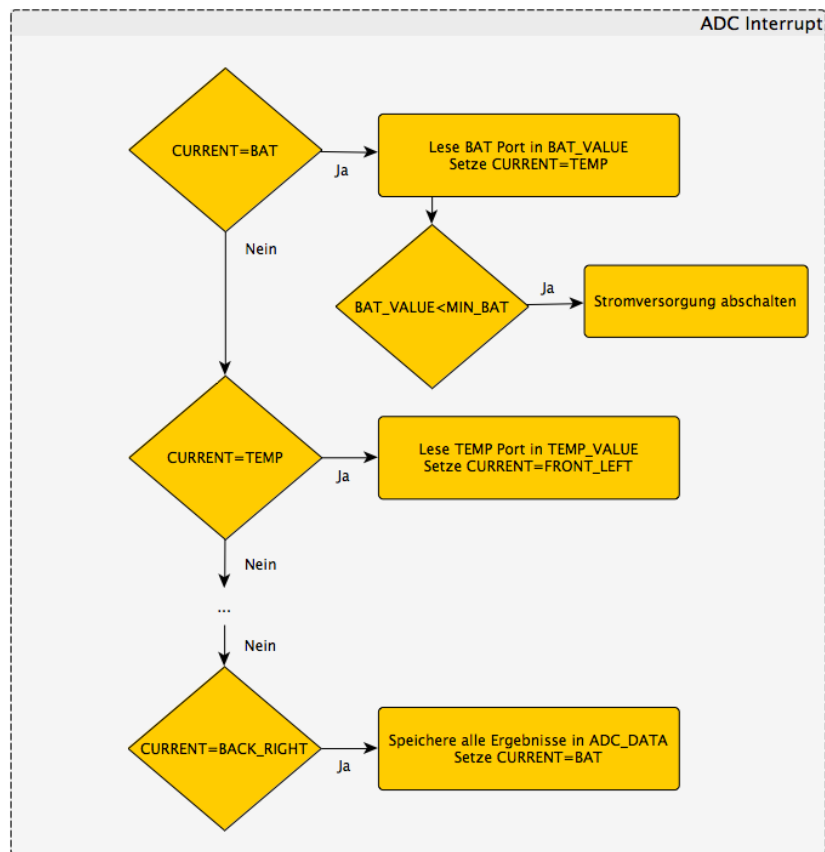


Abbildung 21 ADC-Interrupt

Hauptprogramm

Das Hauptprogramm ist für die Kommunikation mit dem IO-Modul verantwortlich. Hierfür wurde eine freie Bus-Adresse gewählt, unter welcher dieses Modul angesteuert werden kann. Im Hauptprogramm wird endlos auf eine Leseanforderung für diese Adresse gewartet und die gesammelten Sensordaten in einem zusammengefassten Datenpaket übertragen. Dieses setzt sich aus 6x16 Bit breiten Integer Werten der Spannungsmessungen und 8x16 Bit breiten Integer Werten der Ultraschalllaufzeiten zusammen. Die 16 Bit Breite für jeden Sensorwert wurde gewählt, um im Hauptprogramm keine aufwändigen Bitoperationen durchführen zu müssen. Das gesamte Datenpaket ist demnach 224 Bit breit und benötigt für die Übertragung bei einem Durchsatz von 400 kbit/s ca. 560µs. Da die Ultraschallsensoren ca. zwei und die Infrarotsensoren maximal 25 Messungen pro Sekunde durchführen können, ergibt sich somit eine maximale Busauslastung durch das Sensormodul von unter 2%.

5.9.3 IO-Modul

Das IO-Modul sammelt die Sensordaten vom Lagesensor, der Odometrie, des Kompasses sowie vom Sensormodul und stellt diese dem Systemmodul per RS-232 Schnittstelle bereit. Hierzu dient das IO-Modul als Master des i2c Busses. Des Weiteren ist das IO-Modul für die Ansteuerung der Motoren zuständig. Die Software des IO-Moduls ist hierzu in drei voneinander getrennten Prozessen implementiert. Der Abruf und die Auswertung der Sensordaten geschieht über eine zeitgesteuerte Interrupt-Service-Routine. Die Steuerung der Motoren und die Übertragung der gesammelten Sensordaten an das IO-Modul erfolgt im Hauptprogramm. Zur Erfassung der Odometriedaten wird ein Flankenwechselinterrupt genutzt. Die Entwickelte Software für das IO-Modul ist in der beigefügten CD unter „source/odroid-io“ zu finden.

Sensordaten

Zu Beginn werden die Skalen des Gyroskops, des Accelerometers und des Kompasses konfiguriert. Da für die Sensordatenfusion des Gyroskops und des Accelerometers das Zeitdelta zwischen den Abrufen der Sensordaten erforderlich ist, wird die Sensordatenfusion direkt auf dem Mikrokontroller des IO-Modules durchgeführt. Deshalb sind hier die Offsets für den Ausgleich der Nulllage notwendig. Hierzu wird beim Start des Systems von einer Nulllage ausgegangen und die Offsets bestimmt. Nachdem alle Sensoren konfiguriert wurden, wird ein Timer gestartet, welcher alle 10ms eine Interrupt-Service-Routine ausführt. In dieser werden als erstes die Sensordaten des Gyroskops und des Accelerometers abgerufen und der Komplementärfilter mit dem bekannten Zeitdelta von 10ms berechnet. Hiernach werden alle anderen Sensoren abgefragt und die Ergebnisse zusammen mit den Roll und Pitch Winkeln aus der Sensordatenfusion sowie den Odometriedaten in einen global verfügbaren Speicherbereich übertragen. Dieser Prozess läuft endlos ab und stellt dem Hauptprogramm die Sensordaten als zusammengefasstes Datenpaket zur Verfügung.

Odometrie

Die Odometriedaten werden in einer Interrupt-Service-Routine berechnet. Diese wird ausgeführt, sobald eine Zustandsänderung an einem der beiden Pins der Odometriesensoren auftritt. Durch Speichern der vorherigen Zustände der Pins wird bestimmt, welcher der Pins sich verändert hat und der dazugehörige 64Bit breite Integer Wert entsprechend der Laufrichtung in- bzw. dekrementiert. Aufgrund des Radumfangs ergibt sich für jeden Flankenwechsel eine zurückgelegte Strecke von ca. 0,2mm. Hieraus resultiert eine Strecke von mehr als drei Billionen Kilometer

bevor ein 64 Bit breiter Integer Wert überläuft und so die Odometriedaten ungültig werden. Die Odometriedaten werden über einen global verfügbaren Speicherbereich dem Sensordaten-Prozess zur Verfügung gestellt.

Hauptprogramm

Im Hauptprogramm wird auf ein Kommando des Systemmoduls per RS-232 Schnittstelle gewartet. Die möglichen Kommandos sind:

1. Lesen der Sensordaten
2. Vorwärts fahren
3. Rückwärts fahren
4. Links drehen
5. Rechts drehen
6. Antrieb stoppen

Die Sensordaten werden als komplettes Datenpaket übertragen. Dies umfasst das vom Sensormodul übertragene Paket, die drei 16Bit breiten Werte des Kompasses, die 16Bit breiten Roll und Pitch Winkel aus der Sensordatenfusion, die zwei 64Bit breiten Odometriedaten und einen 8Bit breiten Zustand des Antriebs. Hieraus ergibt sich eine gesamt Breite des Sensordatenpakets von 440Bit. Bei einer Übertragungsrates von 128kbit/s der RS-232 Schnittstelle ist demnach eine maximale Abruffrequenz von ca. 290Hz möglich. Diese liegt weit über den 100Hz, mit denen die Sensordaten erfasst werden, und stellt somit kein Problem dar. Die Kommandos für den Antrieb werden direkt beim Eintreffen durch das Hauptprogramm ausgeführt. Hierzu werden zunächst die PWM Kanäle für die Enable Eingänge des Motortreibers und damit auch der Antrieb gestoppt. Anschließend werden die Pins des Motortreibers entsprechend der gewünschten Richtung geschaltet und die PWM Kanäle wieder gestartet. Die Geschwindigkeit der PWM Kanäle wurde zunächst fest gewählt, so dass auch bei einem vollen Akku die 6V Motorspannung nicht überschritten wird.

5.9.4 Sensormodul

Auf dem Systemmodul wurde das Betriebssystem Ubuntu 13.10 installiert und eine WLAN Verbindung konfiguriert. Nach dem Start des Betriebssystems kann per SSH auf das Systemmodul zugegriffen werden. Um die Funktionen des Roboters zu testen, wurde auf dem Systemmodul ein Telemetriemonitor für die Konsole entwickelt. Dies ist auf der beigefügten CD unter „source/odroid/tools/rtop“ zu finden. Das Programm ruft die Sensordaten vom IO-Modul ab und führt die notwendigen Berechnungen und Umrechnungen der Ultraschall- und Infrarotdaten

sowie der zurückgelegten Strecke anhand der Odometriedaten in Millimeter durch. Des Weiteren werden der Kurs, anhand der Kompassdaten, die Temperatur und der Akkustand berechnet. Zum Test des Antriebs ist das Programm in der Lage, diesen anhand von Tastatureingaben zu steuern. Außerdem berechnet das Programm anhand des Kurses und der Odometriedaten eine Position ausgehend vom Startpunkt in Millimeter. Diese wird in einem Grid der Größe von 50x50 Zellen und einer Zellenauflösung von jeweils 100x100mm zusammen mit allen anderen Sensordaten dargestellt. Die Ausgabe des Programmes ist in Abbildung 22 beispielhaft dargestellt.

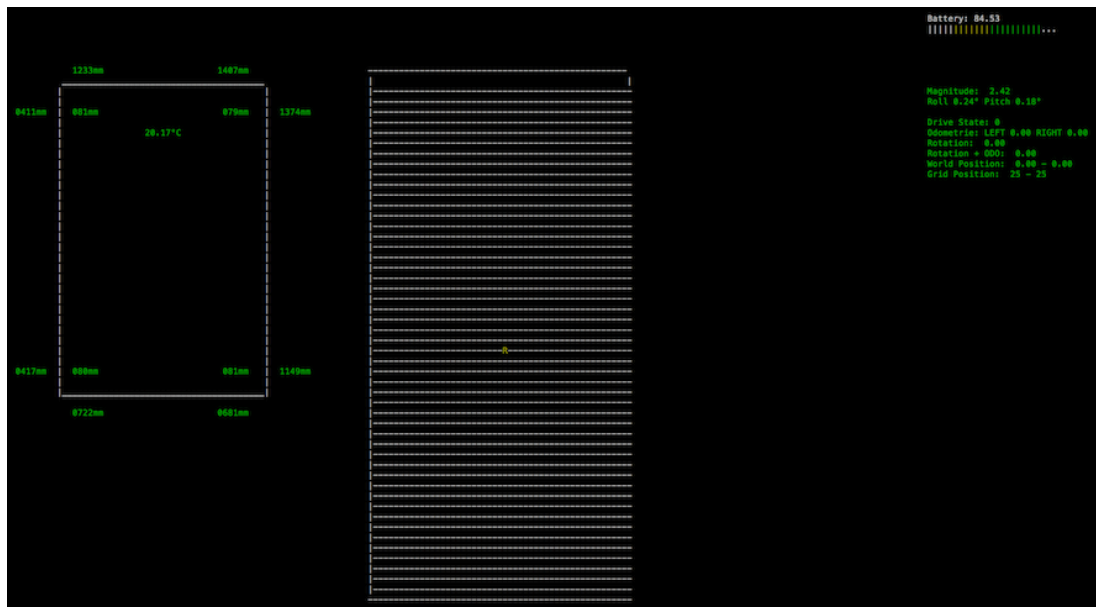


Abbildung 22 rtop Ausgabe

Die Berechnung der Position erfolgt dabei mit Hilfe des Kurses und der zurückgelegten Strecke durch integrieren. Hierzu dient folgende Formel:

$$wp = \begin{pmatrix} x_{alt} + d * \sin \alpha \\ y_{alt} + d * \cos \alpha \end{pmatrix} \quad (17)$$

Hier stellt x_{alt} und y_{alt} die zuvor berechnete Position dar, d gibt die laut Odometrie zurückgelegte Strecke und α den vom Kompass angegebenen Kurs an.

Ein kurzer Test, bei dem eine Strecke von ca. 2m zurückgelegt und anschließend vermessen wurde, zeigte die korrekte Funktion der Positionsberechnung. Hierbei wurde eine Genauigkeit der Position von ca. $\pm 5\%$ erreicht. Ein Test der Infrarotsensoren zeigte, für den relevanten Bereich, eine Genauigkeit von $\pm 2mm$. Der Lagesensor wurde durch leichtes drehen und neigen des Prototypen getestet

und zeigte dabei plausible Werte. Durch horizontales beschleunigen des Prototypen wurde die korrekte Arbeitsweise des Komplementärfilters überprüft. Bei der Überprüfung der Ultraschallsensoren fielen hohe Ungenauigkeiten auf. Unterhalb von 50cm lag die Genauigkeit aller Ultraschallsensoren bei ca. $\pm 5mm$ stieg danach jedoch stark an. Abschließend konnte hierfür keine Ursache gefunden werden. Es wird daher von einem Qualitätsproblem der verwendeten Sensoren ausgegangen.

In dem Testprogramm wurde auch ein einfaches reaktives System, das automatisch Hindernissen und Senken ausweicht, implementiert. So konnte erfolgreich das Zusammenspiel der Sensoren getestet werden.

5.10 Zusammenfassung und Kosten

Zusammenfassend wurde erfolgreich ein fahrtüchtiger Prototyp aufgebaut und die Funktionen mit Hilfe eines Testprogrammes überprüft. Eine mögliche weitere Verbesserung ist eine getrennte Geschwindigkeitssteuerung des Antriebs, um eine möglichst gerade Fahrt zu erreichen. Weiterhin kann im Nachgang dieser Arbeit eine Tilt-Kompensation des Kurses mit Hilfe des Lagesensors umgesetzt werden, um auch beim Überfahren von Hindernissen einen korrekten Kurs zu erhalten. Da die verbauten Ultraschallsensoren nur bis hin zu einer Entfernung von 50cm sinnvoll genutzt werden können ist eine Prüfung anderer Sensoren und ggf. ein Tausch der hier verbauten Sensoren zu empfehlen.

Der fertig aufgebaute Prototyp ist in Abbildung 24 gezeigt.

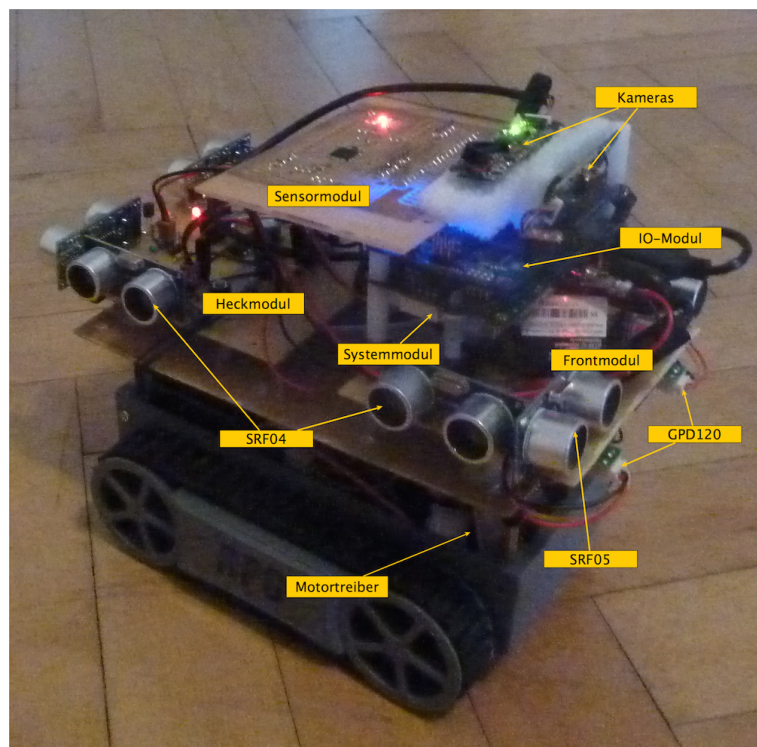


Abbildung 23 Aufgebauter Prototyp

Die Anforderungen an die maximalen Gesamtkosten des Prototype wurden um 1,70 € verfehlt. Eine Aufstellung der Kosten ist in Tabelle 7 aufgeführt.

Bauteil	Anzahl	Einzelpreis	Gesamt
SRF04	4	3 €	12 €
SRF04	4	6 €	24 €
GPD120	4	6 €	24 €
L293D	1	1,20 €	1,20 €
LM1084-it5	1	2 €	2 €
MPU6050	1	5 €	5 €
HMC8583L	1	3 €	3 €
Tiny2313	2	0,95 €	1,90 €
Mega8	1	1,60 €	1,60 €
Odroid U3	1	70 €	70 €
Odroid-IO	1	23 €	23 €
RP6 Chassis	1	41 €	41 €
Kameras	2	17 €	34 €
WLAN Stick	1	7 €	7 €
Odometrie Encoder	2	18,50 €	37 €
Sonstiges Material	1	15 €	15 €
Gesamt			301,70 €

Tabelle 7 Kostenaufstellung

6 Positionsbestimmung

Eine Anforderung an das Verfahren zur Positionsbestimmung des Roboters ist, dass diese auch nach einem Eingriff des Nutzers noch möglich ist. Dies wird auch als das Kidnapped Robot Problem bezeichnet. Bei Kleinstrobotern im Indoor-Betrieb wird häufig auf eine indirekte Positionsbestimmung mittels Odometrie und Korrektur des Schleppfehlers durch SLAM Verfahren gesetzt. Da bei diesen Verfahren die Position jedoch inkrementell bestimmt wird, ist eine Lösung des Kidnapped Robot Problems so nicht möglich. Eine mögliche Lösung wäre der Einsatz eines direkten Ortungssystems wie beispielsweise GPS im Outdoor Bereich. Im Indoor-Bereich sind hierzu Verfahren bekannt, die auf Basis von WLAN oder RFID Funksignalstärken eine Abschätzung der Position vornehmen können. Diese erreichen jedoch aufgrund von Signaldämpfungen durch Hindernisse im Raum meist nur eine geringe Genauigkeit. Des Weiteren haben solche Verfahren zumeist einen relativ hohen Installationsaufwand und sollen aufgrund der Anforderung, auf aufwändige externe Ortungssysteme zu verzichten, im Rahmen dieser Arbeit nicht untersucht werden. Daher wird im Folgenden ein Verfahren vorgeschlagen und untersucht, welches mit Hilfe von Odometriedaten eine Positionsbestimmung durchführt und anhand von festen visuell zu erkennenden Referenzpunkten eine Positionsbestimmung auch nach einem Nutzereingriff ermöglichen soll. (Klein)

6.1 Odometrie Referenzpunktkorrektur

Zur inkrementellen Bestimmung der Position können Odometrie- oder Geschwindigkeits- und Winkelsensoren genutzt werden. Ein entscheidender Nachteil dieses Verfahrens liegt in der Genauigkeit der zur Verfügung stehenden Sensoren. So ist eine hundertprozentige Genauigkeit bei der Bestimmung von Geschwindigkeit und Kurs nicht möglich. Aufgrund der inkrementellen Arbeitsweise des Verfahrens resultiert hieraus ein sich immer weiter aufsummierender Fehler. Zur Veranschaulichung des inkrementellen Fehlers soll hierzu Abbildung XX dienen. So zeigt die blaue Linie die tatsächlich abgefahrene Strecke von ca. 18m Länge und die rote Linie die inkrementell bestimmte Position anhand von Geschwindigkeit und Kurs. Es handelt sich in diesem Fall um eine rechnerische Worst-Case Betrachtung bei einer Genauigkeit für die Geschwindigkeit von $\pm 1\%$ und für den Kurs von $\pm 2^\circ$.

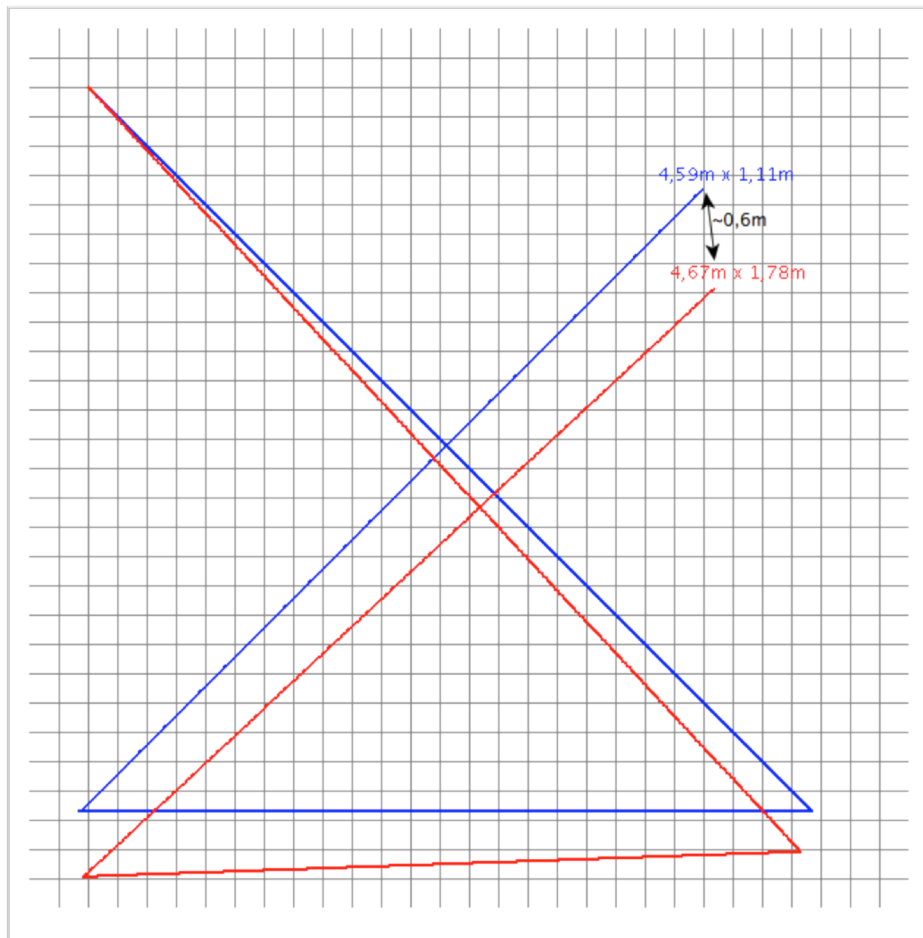


Abbildung 24 Aufsummierender Fehler

Wie in Abbildung 25 zu sehen liegt die berechnete Position am Ende der Fahrt ca. 0,6m neben der tatsächlichen Position. Eine mögliche Lösung dieses Problems ist die Korrektur der Position anhand von Referenzpunkten. Hierzu werden während der Fahrt wiedererkennbare Referenzpunkte gesucht und mit der bis hierhin errechneten Position gespeichert.

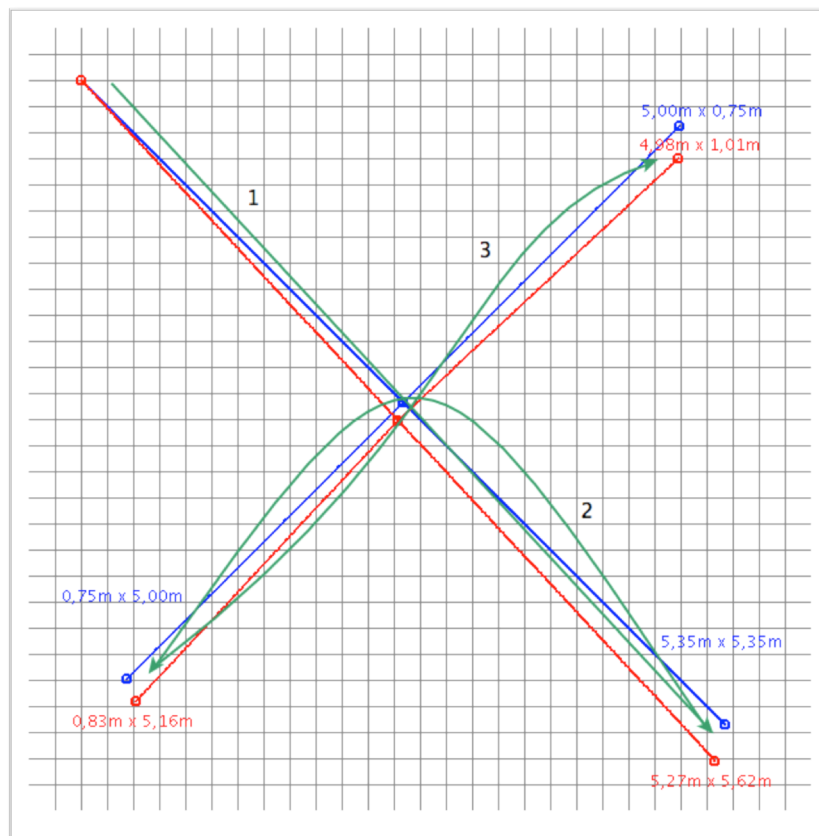


Abbildung 25 Referenzpunktkorrektur

Wird im späteren Verlauf ein Referenzpunkt wiedererkannt, kann die aktuelle Position anhand der gespeicherten Position des Referenzpunktes korrigiert werden. Wie in Abbildung 26 zu sehen könnte der Roboter so beispielweise zu Beginn in einem 45° Kurs von der oberen in die untere Ecke fahren (1). Auf dem Weg dorthin wird in der Mitte ein Referenzpunkt erkannt. Im zweiten Schritt kann der Roboter nun immer wieder von der Mitte aus in die anderen Ecken fahren (2. und 3.). Somit reduziert sich der inkrementelle Fehler zu jeder Ecke, auf den Fehler der bei einer direkten Anfahrt dieser Ecke und der Abweichung des Referenzpunktverfahrens auftritt. Ist die Abweichung des Referenzpunktverfahrens klein genug, ließe sich so in dem in Abbildung 25 und 26 skizzierten Beispiel der Fehler bei der linken unteren und rechten oberen Position gegenüber der in Abbildung 25 gezeigten einfachen Positionsbestimmung mehr als halbieren. Ein Beispiel für ein solches Vorgehen ist das CV-SLAM Verfahren. Hierbei werden mittels Computer Vision Verfahren an der Raumdecke Referenzpunkte gesucht und gespeichert. Werden diese später wiedererkannt, kann eine Korrektur der Position vorgenommen werden. Ein weiteres Beispiel ist das RatSLAM verfahren. Vereinfacht wird hierbei ein Computermodell des Hippocampus einer Ratte und eine Kamera genutzt, um

Referenzpunkte im Raum zu erkennen. Der Hippocampus dient hierbei als Gedächtnis für die visuelle Objekterkennung und liefert anhand der Kamerabilder Referenzpunkte und erkennt diese auch wieder. (Ohne, ratslam) (Chang, 2012)

6.2 Positionsbestimmung mittels Landmarke

Da im Rahmen dieser Arbeit eine absolute Positionsbestimmung gefordert ist und hierbei auf aufwändige und kostenintensive Ortungssysteme verzichtet werden soll, wird im Folgenden ein kombiniertes Verfahren aus Odometrie und visuelle erkennbarer Landmarken vorgeschlagen und untersucht. Hierbei wird eine eindeutig erkennbare Landmarke an der Raumdecke angebracht, welche als

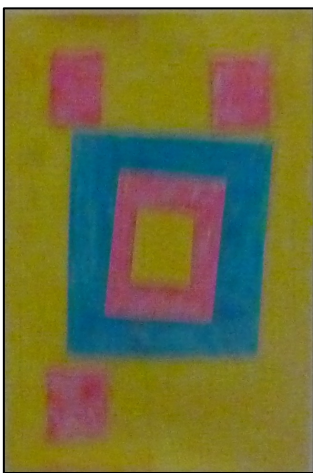


Abbildung 26 Landmarke

Referenzpunkt dienen soll. Die Raumdecke wurde hierbei gewählt, da hier von einer hohen Wahrscheinlichkeit ausgegangen wird, stets freie Sicht auf die Landmarke zu haben. Zur Vereinfachung der Erkennung wurde hierbei auf ein Farbsegmentierungsverfahren gesetzt. Das heißt die Markierung bzw. Landmarke wird anhand ihrer Farbe und Struktur erkannt. Als Landmarke wird eine mit Neonfarben auf ein Hochglanzpapier gezeichnete Struktur genutzt. Die Neonfarbe wird verwendet, da erwartet wird, dass so eine Farbsegmentierung auch unter verschiedenen Lichtverhältnissen möglich ist. Die gezeichnete Landmarke ist in Abbildung 27 dargestellt.

Um die Landmarke zuverlässig zu erkennen, wird das Kamerabild der Raumdecke in seine gelben, blauen und roten Anteile zerlegt und auf die vorgegebene Struktur untersucht. Zur Skalierung des Verfahrens können somit Landmarken in unterschiedlicher farblicher Kodierungen an der Raumdecke befestigt und eindeutig erkannt werden. Nachfolgend werden die Verfahren zur Farbsegmentierung und Strukturerkennung sowie der Positionsbestimmung vorgestellt. Anschließend wird das Verfahren getestet und bewertet.

6.2.1 Farbsegmentierung anhand des HSV Farbraums

Digitale Kamerasysteme nehmen Bilder typischerweise im RGB Format auf. Hierbei wird ein Pixel mit seinen Rot-, Grün- und Blauanteilen angegeben. Um ein Bild anhand einer Farbe zu filtern, ist der RGB Farbraum jedoch nur bedingt geeignet. So ist eine bestimmte Farbe bei unterschiedlicher Helligkeit nicht über einfache Schwellwertfunktion zu erkennen. Eine Lösung ist der HSV Farbraum. Bei diesem wird ein Pixel mit seinem Farbwert (hue), seiner Sättigung (saturation) und seiner Helligkeit (value) angegeben. Der Farbwert wird dabei als Winkel auf einem

Farbkreis angegeben. Die Werte für die Sättigung und Helligkeit werden wiederum zwischen 0 und 100% bzw. 0 und 1 angegeben. Zur Umrechnung eines RGB Pixels, für das gilt $R, G, B \in [0,1]$, in den HSV Farbraum dienen folgende Gleichungen: (Ohne, HSV-Farbraum)

$$MAX := \max(R, G, B), MIN = \min(R, G, B) \quad (18)$$

$$H := \begin{cases} 0, & \text{falls } MAX = MIN \\ 60^\circ * \left(0 + \frac{G - B}{MAX - MIN}\right), & \text{falls } MAX = R \\ 60^\circ * \left(2 + \frac{B - R}{MAX - MIN}\right), & \text{falls } MAX = G \\ 60^\circ * \left(4 + \frac{R - G}{MAX - MIN}\right), & \text{falls } MAX = B \end{cases} \quad (19)$$

$$\text{falls } H < 0^\circ \text{ dann } H := H + 360^\circ \quad (20)$$

$$S := \begin{cases} 0, & \text{falls } MAX = 0 \\ \frac{MAX - MIN}{MAX}, & \text{sonst} \end{cases} \quad (21)$$

$$V := MAX \quad (22)$$

Ist ein Bild in den HSV Farbraum konvertiert, können die Schwellwerte für die Farbsegmentierung der Landmarke experimentell bestimmt werden. Sind die Schwellwerte bekannt, kann das Kamerabild entsprechend gefiltert und anschließend nach den Strukturen gesucht werden.

6.2.2 Sobel

Beim Sobel Algorithmus handelt es sich um ein Kanten Detektions Verfahren. Solch ein Algorithmus wird beispielsweise von vielen Verfahren zur Feature Detektion als Vorverarbeitung der Bilddaten benötigt. So können über Kanten und deren Richtung beispielsweise Ecken identifiziert werden. Der Sobel Algorithmus ist einer der bekanntesten Vertreter der Kanten Detektions Verfahren und soll hier daher vorgestellt werden.

Kanten sind ein optisch prägnantes Bildmerkmal und umreißen beispielsweise Objekte und Menschen. So können Kanten zur Abgrenzung von Objekten dienen und spielen daher auch beim menschlichen Sehen eine wichtige Rolle. Als Kante sind im Folgenden Stellen definiert, welche eine abrupte Helligkeitsänderung zu

ihrer Umgebung aufweisen. Anhand dieser Definition lässt sich erkennen, dass ein Kanten Detektions Verfahren mittels einer Matrix und Faltungsoperation umgesetzt werden kann. So kann eine Matrix definiert werden, welche über ein Helligkeitsbild geschoben wird und für jeden Pixel die Umgebung so gewichtet, dass bei relativ gleichmäßiger Helligkeit eine Nullsumme entsteht. Verändert sich allerdings die Helligkeit in der Umgebung des betrachteten Pixels, so erhöht sich auch die Ausgabe dieser Matrix. Genau dies ist die Vorgehensweise des Sobel Algorithmus. Hierzu können folgende zwei Matrizen genutzt werden (Wagner, 2006):

$$S_x = \begin{pmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{pmatrix} \quad (23)$$

$$S_y = \begin{pmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{pmatrix} \quad (24)$$

Zur Faltung werden die einzelnen Umgebungspixel mit den korrespondierenden Elementen der Matrix multipliziert. Die Summen der einzelnen Produkte bilden die Ausgabe für den betrachteten Pixel und liefern ein Bild der Kanten. Hierbei liefert S_x alle vertikalen und S_y alle horizontalen Kanten. Da für die Berechnung der Kanten Helligkeitsinformationen notwendig sind, müssen diese bei Farbbildern zuvor, beispielsweise durch Konvertierung in ein Graustufenbild, bestimmt werden. Zur Kombination der so gewonnen vertikalen und horizontalen Kantenbilder kann folgende Formel dienen (Wagner, 2006):

$$G_{i,j} = \sqrt{G_{x_{i,j}}^2 + G_{y_{i,j}}^2} \quad (25)$$

Wobei $G_{x_{i,j}}$ das Ergebnis, auch Gradientenbild genannt, der Faltung mit S_x an Position i,j und $G_{y_{i,j}}$ entsprechend mit S_y ist. In G sind hiernach sowohl die vertikalen als auch horizontalen Kanten enthalten. Beispielhaft ist das Ergebnis der Anwendung des kombinierten Sobel Verfahrens in Abbildung 29 dargestellt. Als Eingabe des Verfahrens diente hierbei das in Abbildung 28 gezeigte Bild. (Wagner, 2006)

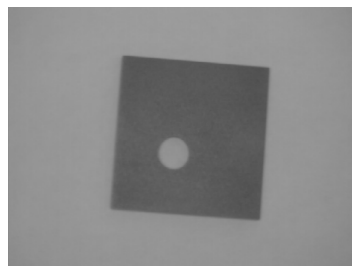


Abbildung 27 Sobel Eingabe
(Wagner, 2006)

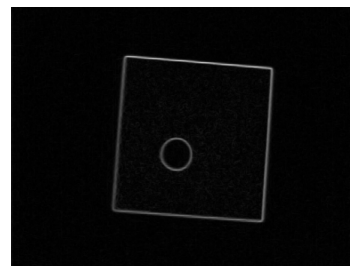


Abbildung 28 Sobel Ausgabe
(Wagner, 2006)

6.2.3 Kantenverfolgung / Strukturanalyse

Um in einem Bild Strukturen zu identifizieren, kann eine Kantenverfolgung und topologische Strukturanalyse durchgeführt werden. Hierzu wird zunächst ein Gradientenbild des zu untersuchenden Bildes, beispielsweise mit dem Sobel Verfahren, erzeugt. Anschließend werden die so gefundenen Kanten verfolgt bzw. verbunden. Zur Verfolgung der Kanten wird wie folgt vorgegangen:

1. Bestimme einen Kantenpunkt im Gradientenbild
2. Suche den nächsten Kantenpunkt senkrecht zur aktuellen Gradientenrichtung
3. Ein Kantenpunkt ist gefunden wenn
 - dieser in hinreichender Entfernung zum aktuellen ist
 - eine hinreichend geringe Richtungsabweichung aufweist
 - ein hinreichend starker Gradient existiert

Hierbei wird das Entfernungskriterium zur Überbrückung von Lücken im Kantenverlauf genutzt. Das Richtungskriterium dient der Selektierung der korrekten Kante, beispielsweise bei Kreuzungen von Kanten, und steigert die Unempfindlichkeit gegenüber Rauschen. Das Kantenstärkenkriterium wird ebenfalls zur Steigerung der Unempfindlichkeit gegenüber Rauschen genutzt. (Pohle) Die so gefundenen Kantenverläufe können anschließend durch eine topologische Strukturanalyse in zusammenhängende Strukturen verbunden werden. Siehe hierzu „Topological Structural Analysis of Digitized Binary Images by Border Following“. (Satoshi Suzuki, 1985)

6.2.4 Erkennen der Landmarke und ihres Rotationswinkels

Zur Erkennung der Landmarke werden zunächst geeignete Schwellwerte für den gelben, blauen und roten Teil der Landmarke experimentell bestimmt. Mit Hilfe der Schwellwerte und Farbsegmentierung kann nun das Eingangsbild der Kamera in Bilder mit den entsprechenden Farbanteilen zerlegt werden. Für jedes dieser Bilder wird wiederum ein Gradientenbild erzeugt und mit Hilfe der Kantenverfolgung und Strukturanalyse eine Liste mit Konturen erzeugt. Die so erzeugten Listen von Konturen für die gelben, blauen und roten Anteile können nun nach Rechtecken, welche die durch die Landmarke vorgegebene Verschachtelung aufweisen, durchsucht werden. Zum Ausgleich der Perspektive zwischen zwei erkannten Landmarken ist die Bestimmung des Winkels der Landmarke erforderlich. Hierzu werden zwei feste Punkte benötigt, die unabhängig von der Perspektive eindeutig bestimmt werden können. Daher sind die drei roten äußeren Rechtecke auf der Landmarke in einem Dreieck angeordnet. Zur Bestimmung des Winkels werden als

erstes die Mittelpunkte der drei äußeren roten Rechtecke bestimmt. Indem die Distanz in Pixel zwischen diesen drei Punkten berechnet wird, kann der in Abbildung 27 linke untere Punkt, im Folgenden Punkt C genannt, eindeutig identifiziert werden. Hierzu werden die Punkte mit der kürzesten Distanz bestimmt und somit per Ausschlussverfahren der Punkt C gefunden. Da es sich um ein rechtwinkliges Dreieck handelt, kann nun mit Hilfe des Kosinussatzes der Winkel zwischen dem Punkt C und einem der verbliebenen zwei Punkte bestimmt werden. Zum Ausgleich von Ungenauigkeiten der Punktpositionen kann ein experimentell bestimmter Schwellwert genutzt werden, um die beiden verbliebenen Punkte eindeutig zu identifizieren. Da so alle drei Punkte unabhängig von der Perspektive erkannt werden, können nachfolgend auch zwei eindeutige Punkte gewählt und der Winkel zwischen diesen bestimmt werden. (Ohne, Kosinussatz)

Für den Ausgleich der unterschiedlichen Perspektiven kann folgende Formel genutzt werden:

$$p_{korrigiert} = \begin{pmatrix} x_e + ((x_m - x_e) * \cos \alpha - (y_m - y_e) * \sin \alpha) \\ y_e + ((x_m - x_e) * \sin \alpha - (y_m - y_e) * \cos \alpha) \end{pmatrix} \quad (26)$$

Wobei gilt:

$x_e = x$ Position des erkannten Punktes

$y_e = y$ Position des erkannten Punktes

$x_m = x$ Position der Bildmitte

$y_m = y$ Position der Bildmitte

$\alpha =$ Winkeldifferenz zwischen Key – Frame und erkannter Landmarke

6.2.5 Positionsbestimmung

Für das hier vorgeschlagene Verfahren ist ein quadratisches Eingabebild erforderlich. Aufgrund des Seitenverhältnisses der eingesetzten Kamera wird hierzu ein Quadrat von 600x600 Pixel ausgehend von der Mitte des Bildes ausgeschnitten und nachfolgend als Eingabebild betrachtet. Da die Landmarke eindeutig erkannt werden kann, ist das Verfahren zur Positionsbestimmung relativ einfach. Wird eine Landmarke erkannt, die zuvor noch nicht gesehen wurde, wird diese zusammen mit der aktuellen per Odometrie bestimmten Position als Key-Frame gespeichert. Ein Key-Frame beinhaltet dabei folgende Daten:

1. Eindeutige Identifikationsnummer
2. Pixel Position des Punktes C
3. Rotationswinkel der Landmarke
4. Position laut Odometrie
5. Koeffizient

Eine eindeutige Identifikationsnummer kann durch farbliche Kodierung der Landmarke festgelegt werden. Die Identifikationsnummer ist hierbei erforderlich, um Landmarken eindeutig vergleichen zu können. So ist es nur möglich, eine absolute Position zu bestimmen, wenn die aktuell gesehene Landmarke mit der identischen zuvor gespeicherten Landmarke verglichen wird. Dies macht auch deutlich, dass in einem Raum nur unterschiedlich kodierte Landmarken angebracht werden dürfen. Da sowohl die Odometrie als auch die Sensoren zur Hinderniserkennung eine Entfernung in Millimetern liefern, soll das Verfahren der Positionsbestimmung ebenfalls in diesem Maßstab arbeiten. Aufgrund der Tatsache, dass die Entfernungen zwischen zwei erkannten Landmarken nur in Pixel berechnet werden können, ist hier jedoch eine Umrechnung erforderlich. Hierzu wird vorgeschlagen, dass nachdem ein Key-Frame gespeichert wurde, während der Weiterfahrt des Roboters mit Hilfe der Odometriedaten ein Koeffizient zur Umrechnung bestimmt und ebenfalls gespeichert wird. Ist nun eine bekannte Landmarke, zu welcher auch ein Koeffizient gespeichert wurde, in Sicht, kann auch die Position des Roboters anhand der Landmarke bestimmt werden. Hierfür wird zunächst ein eindeutiger Punkt auf der Landmarke identifiziert. Dies soll im Folgenden der in Kapitel 6.2.4 beschriebene Punkt C sein. Zum Ausgleich der möglichen unterschiedlichen Perspektiven zwischen gespeicherter und aktuell gesehener Landmarke wird der Punkt C zunächst soweit um den Mittelpunkt des Bildes gedreht, sodass eine ähnlich Perspektive entsteht. Hierzu kann die Differenz zwischen den gemessenen Winkeln der aktuell gesehenen Landmarke und des Key-Frames genutzt werden. Nun können die Entfernungen in x und y Richtung zwischen den Punkten in der gespeicherten und der aktuell gesehen Landmarke in Pixel berechnet und anschließend mit Hilfe des Koeffizienten umgerechnet werden. Die Position des Roboters ergibt sich nun durch Verschieben der zuvor gespeicherten Position des Key-Frames um die berechneten Entfernungen. Nachfolgend kann so die aktuell per Odometrie bestimmte Position wie Eingangs beschrieben ggf. korrigiert werden. Des Weiteren kann nach erkennen eines Nutzereingriffs eine zuvor gespeicherte Landmarke gesucht und damit die Position neu bestimmt werden. Eine Einschränkung des Verfahrens ist hierbei jedoch, dass eine absolute Positionsbestimmung nach einem Nutzereingriff nur möglich ist, wenn zuvor bereits eine Landmarke gesehen und gespeichert wurde.

Die Position des Roboters kann somit mit folgender Formel berechnet werden:

$$wp = wp_{key_i} + \left((p_{key_i} - p_{korrigiert_i}) * k_i \right) \quad (27)$$

Hier gilt:

wp_{key_i} = Weltposition des Key – Frames mit id i (per Odometrie bestimmt)

p_{key_i} = Pixel Position des Key – Frames mit id i

k_i = Koeffizient zur Umrechnung des Key – Frames mit id i

6.2.6 Tests

Aufgrund der eindeutigen Identifizierung der Landmarke wird davon ausgegangen, dass sich das Verfahren einfach skalieren lässt. Daher wird für den folgenden Test das Verfahren nur mit einer Landmarke betrachtet. Für den Test wurde ein Programm auf Basis von opencv entwickelt und direkt auf dem Prototypen ausgeführt. Dieses ist auf der beigefügten CD unter „source/odroid/ceil/test“ zu finden. Vor Beginn des Tests wurde eine Position vermessen und am Boden markiert. Außerdem wurde durch Vergleichsmessungen manuell ein Koeffizient bestimmt. Der Prototyp wurde zu Beginn des Tests auf den markierten Punkt gesetzt und das Programm mit der bekannten Position und dem bekannten Koeffizienten gestartet. Zu Beginn wird durch das Programm eine Landmarke gesucht und als Key-Frame gespeichert. Anschließend wird fortwährend nach der Landmarke gesucht und die berechnete Position ausgegeben. Zur Berechnung der Abweichung des Verfahrens wurde die mittlere Position an der Ausgangsposition und an sieben weiteren zuvor vermessenen Punkten rund um die Landmarke bestimmt. Für jedes so entstandene Paar von Position wurde die Entfernung zwischen gemessener und durch das Verfahren bestimmter Position in Millimeter berechnet. Die Abweichungen sind in dem Diagramm in Abbildung 30 und die entsprechenden Entfernungen in Abbildung 31 dargestellt. Hier ist eine Korrelation zwischen Entfernung zur Landmarke und Abweichung zu erkennen. In einem weiteren Test wurde auch eine Korrelation zwischen unterschiedlicher Rotation zwischen Key-Frame und Test-Frame erkannt. Dies ist in dem Diagramm in Abbildung 32 dargestellt. Sowohl die rotations- als auch entfernungsabhängige Abweichung kann auf eine nicht in Waage und damit parallel zur Raumdecke angebrachten Kamera zurückgeführt werden. Dies führt zu einer perspektivischen Verzerrung des Eingangsbildes und somit zu den Abweichungen.

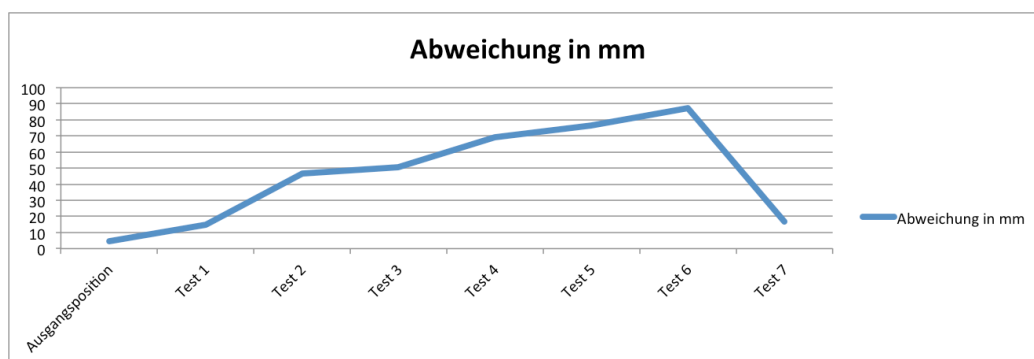


Abbildung 29 Abweichung in mm

Die maximal mögliche Entfernung zum Key-Frame ist bei diesem Verfahren von der Höhe der Raumdecke abhängig. Für den hier durchgeführten Test lag diese bei ca. 70cm rund um die Landmarke. Das heißt für einen Raum von 7x7m Grundfläche könnte mit 25 unterschiedlichen Landmarken eine vollständige Abdeckung erreicht

werden. Da dieses Verfahren jedoch im Normalbetrieb nur zur Korrektur der mittels Odometrie bestimmten Position und lediglich im Falle eines Nutzereingriffes zur direkten Positionsbestimmung genutzt werden soll, ist eine vollständige Abdeckung nicht erforderlich.

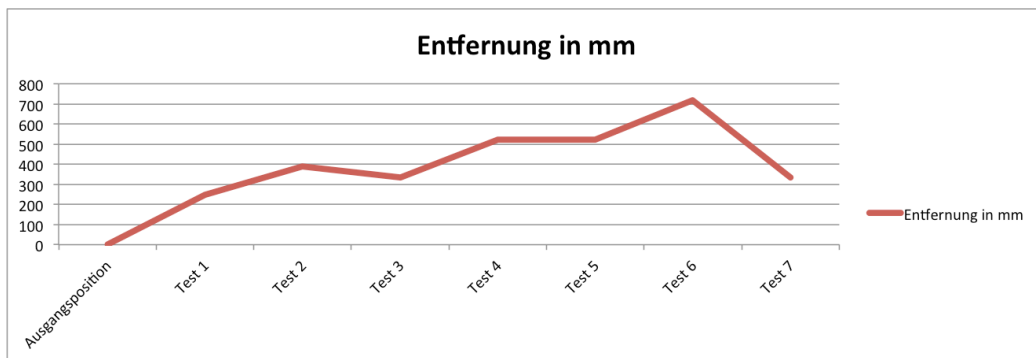


Abbildung 30 Entfernung zum Startpunkt

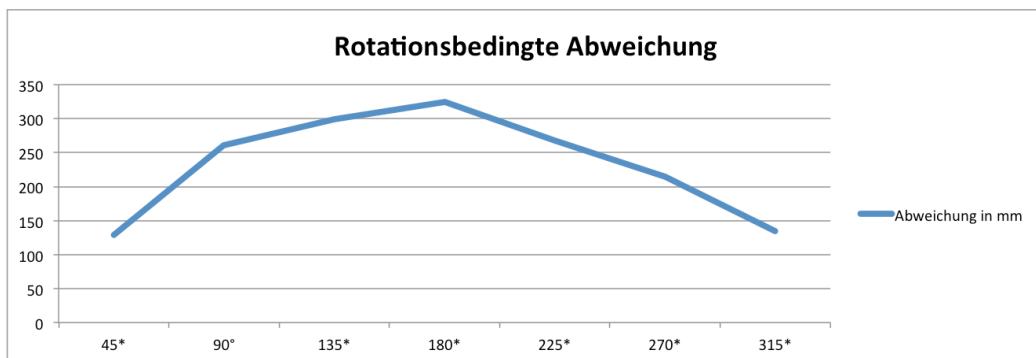


Abbildung 31 Rotationsabhängige Abweichung

Zur Lösung der rotationsbedingten Abweichung wird vorgeschlagen, einen Key-Frame aus unterschiedlichen Perspektiven aufzunehmen. Hierzu könnte sich der Roboter beim Erkennen einer zuvor nicht bekannten Landmarke unter dieser um 360° drehen, um aus so vielen Winkeln wie möglich Landmarken zu erhalten. Zur Positionsbestimmung kann anschließend der Key-Frame mit der geringsten Winkelabweichung genutzt werden. Es wird erwartet, dass die rotationsbedingte Abweichung dadurch keine signifikante Auswirkung mehr auf die Genauigkeit des Verfahrens hat.

Zum Test der Erkennungsrate bei unterschiedlichen Lichtverhältnissen wurde der Raum abgedunkelt und anschließend mit einer relativ lichtarmen Deckenlampe und einer regulierbaren Stehlampe beleuchtet. Der Test wurde bei allen Lichtverhältnissen an der gleichen Position durchgeführt. Zur Bestimmung der Erkennungsrate wurden 200 Bilder der Raumdecke aufgenommen und mit diesen eine Landmarkenerkennung durchgeführt. Die Anzahl erkannter Landmarken wurde

in einen Prozentsatz umgerechnet. Die Ergebnisse sind in Abbildung 33 dargestellt. Hier ist zu sehen, dass auch bei schlechten Lichtverhältnissen noch auf 20% der Bilder eine Landmarke erkannt wird. Da so mindestens in jedem zehnten Bild eine Landmarke erkannt wird und das Verfahren in einem Test eine Framerate von ca. zwölf Bildern pro Sekunde erreichte, ist davon auszugehen, dass dies für das hier vorgestellte Verfahren zur Positionsbestimmung als ausreichend angesehen werden kann.

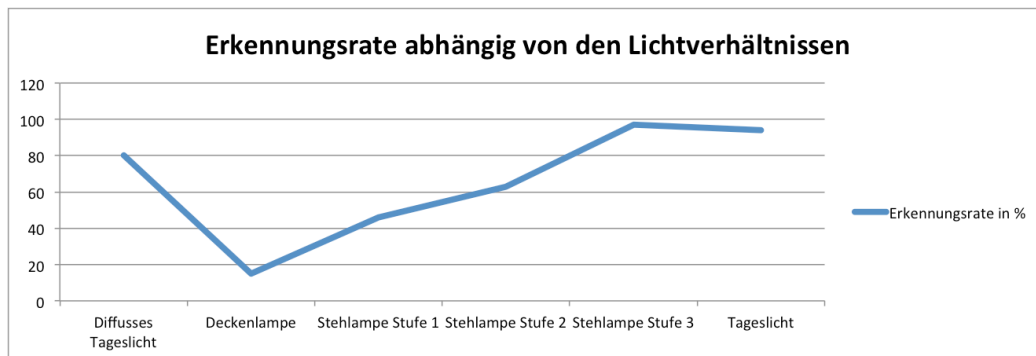


Abbildung 32 Erkennungsrate

6.2.7 Fazit

Unter der Voraussetzung, dass vor einem Nutzereingriff bereits eine Landmarke erkannt wurde, kann als Fazit gezogen werden, dass das hier vorgeschlagene Verfahren zur Positionsbestimmung durchaus dafür geeignet ist, auch nach einem Nutzereingriff eine absolute Positionsbestimmung durchzuführen und so das Kidnapped Robot Problem zu lösen. Um zu gewährleisten, dass bereits vor einem Nutzereingriff eine Landmarke gespeichert wurde, könnte als Bedingung für die Inbetriebnahme des Roboters eine Landmarke in Sicht gestellt werden. Es konnte gezeigt werden, dass für eine mit diesem Verfahren bestimmte Position eine Abweichung von unter 10cm zu erwarten ist. Des Weiteren konnte ein Verfahren zum Ausgleich der rotationsbedingten Abweichung vorgeschlagen werden. Unter der Voraussetzung das eine im Nachgang dieser Arbeit zu Entwickelnde Basisstation beispielweise visuell erkannt werden kann, wird erwartet, dass die mit diesem Verfahren erreichte Genauigkeit ausreicht, um diese auch nach einem Nutzereingriff jederzeit anfahren zu können.

7 Objekterkennung

Um im Nachgang dieser Arbeit einen autonomen Roboter entwickeln zu können, der in der Lage ist die Munition selbständig zu suchen und einzusammeln, werden im Folgenden Verfahren zur visuellen Objekterkennung untersucht und bewertet. Hierbei ist das Ziel eine möglichst zuverlässige Erkennung der in Abbildung 34 gezeigten Munition aus der in Abbildung 33 gezeigten Perspektive zu erreichen.



Abbildung 33 Nerf Munition



Abbildung 34 Nerf Munition Perspektive

7.1.1 Haar-like Features

Haar-like Features dienen der Extraktion von Bildmerkmalen, welche wiederum für die Objekterkennung genutzt werden können. Hierbei werden Helligkeitsunterschiede unterschiedlicher Regionen eines Objektes als Merkmal betrachtet. Ursprünglich wurde dieses Verfahren zur Erkennung von Gesichtern eingesetzt. So sind beispielsweise Augen dunkler als Nase oder Stirn was als Merkmal eines Gesichtes dienen kann. Es hat sich jedoch gezeigt dass mit diesem Verfahren auch andere Objekte erkannt werden können. Als Haar-like Features

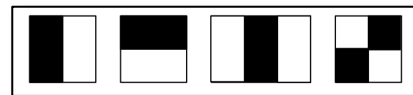


Abbildung 35 Haar-like Features (Blauth, 2012)

dienen in dem ursprünglich von Viola und Jones 2001 vorgestelltem Verfahren die in Abbildung 36 gezeigten Masken. Diese können als eine Fensterfunktion betrachtet werden welche zum Eingrenzung eines Features dienen. Hierbei werden die Grauwerte aller Pixel aufsummiert und jene innerhalb der weißen Bereiche

positiv und in den Schwarzen Bereichen negativ gewichtet. Zur schnellen Berechnung eines Features dient hierbei ein Integral-Bild. Dieses wird wie folgt bestimmt. (Blauth, 2012)

$$ii(x, y) = \sum_{x' \leq x, y' \leq y} i(x', y') \quad (28)$$

Dabei beschreibt $ii(x, y)$ den Pixel an Position (x, y) im Integralbild und $i(x, y)$ entsprechend im Eingangsbild. Die Pixel im Integralbild enthalten somit die Summe der Pixel zwischen x, y und dem Bildursprung des Eingangsbildes. Um die Anzahl zu bildender Summen zu Reduzieren kann das Integralbild auch Rekursiv bestimmt werden. (Blauth, 2012)

$$s(x, y) = s(x, y - 1) + i(x, y) \quad ii(x, y) = ii(x - 1, y) + s(x, y) \quad (29)$$

Hierbei ist s die kumulative Zeilensumme und $s(x, -1) = ii(-1, y) = 0$

Nun kann beispielsweise die Summe der Pixel der in Abbildung 37 gezeigten Fläche F mit Hilfe des Integralbildes wie folgt berechnet werden. (Blauth, 2012)

$$F = D - B - C + A \quad (30)$$

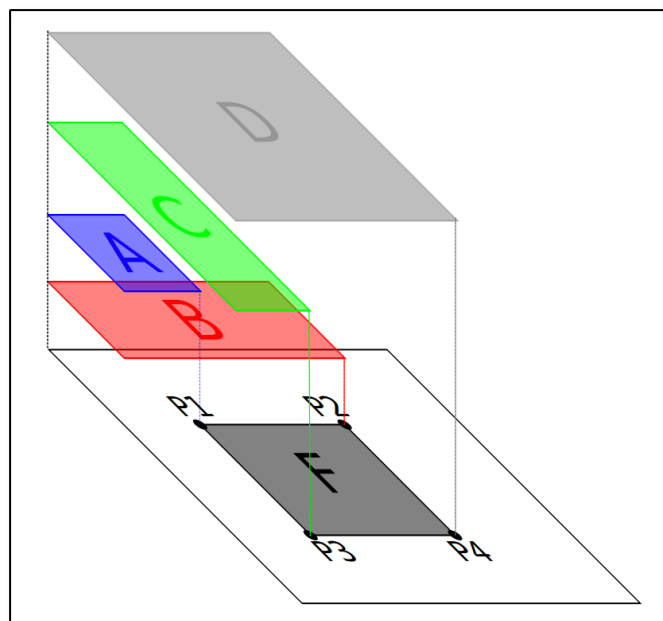


Abbildung 36 Berechnung anhand Integralbild (Blauth, 2012)

Zur Berechnung eines Features können somit die Teilflächen entsprechend der weißen und Schwarzen Bereichen des Features gewählt werden.

Im Jahr 2002 wurden von Lienhart und Maydt die zuvor gezeigten Features erweitert. Hierbei ist die auffälligste Erweiterung die nun vorhanden um 45° geneigten Features. Diese Erweiterung soll dabei bessere Features liefern und somit die Objekterkennung insgesamt verbessern.

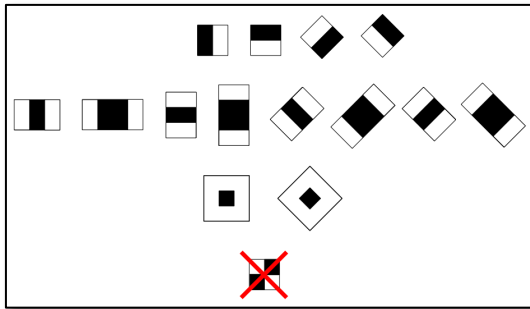


Abbildung 37 Erweiterte Features (Blauth, 2012)

Wie in der Abbildung 38 gezeigten erweiterten Menge an Features zu sehen, wurde dabei auch eines der vorherigen Features entfernt, da dieses durch die neuen ersetzt wurde. Auch diese Features lassen sich wieder durch ein Integralbild effizient berechnen. Das zuvor gezeigte Verfahren gilt dabei jedoch nur für alle

nicht um 45° geneigten Features. Für alle anderen ist ein weiteres Integralbild erforderlich. Dieses sogenannte Rotated Integral Image lässt sich dabei wie folgt berechnen. (Blauth, 2012)

$$rii(x, y) = \sum_{x' \leq x, y' \leq x - |x' - y|} i(x', y') \quad (31)$$

Das sich daraus ergebende Rotierte Integralbild rii lässt sich auch wieder durch ein Rekursives Verfahren effizienter bestimmen. Hierzu wird das Eingangsbild i erst von links nach rechts sowie von oben nach unten und in einen weiteren Durchlauf von rechts nach links sowie unten nach oben durchlaufen. Für den ersten Durchlauf dient dabei folgende Gleichung: (Blauth, 2012)

$$rii(x, y) = rii(x - 1, y - 1) + rii(x - 1, y) + i(x, y) - rii(x - 2, y - 1) \quad (32)$$

Der zweite Durchlauf wird dann mit folgender Gleichung durchgeführt:

$$rii(x, y) = rii(x, y) + rii(x - 1, y + 1) - rii(x - 2, y) \quad (33)$$

Die Summe eines um 45° geneigten Rechteckes $r = (x, y, w, h, 45^\circ)$ kann hiernach mit folgender Gleichung bestimmt werden:

$$RecSum(r) = rii(x + w, y + w) + rii(x - h, y + w) + rii(x - h, y + h) - rii(x, y) - rii(x + w - h, y + w - h) \quad (34)$$

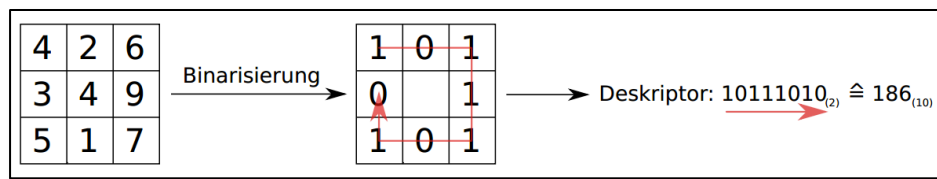


Abbildung 39 LBP Konstruktion (Blauth, 2012)

Hierbei ist g_i der i -te umgebende Pixel von g_m , wobei 0 links oben ist und alle Weiteren im Uhrzeigersinn verlaufen. Es entsteht hierbei ein Deskriptor der beschreibt welche umgebenden Pixel heller oder dunkler im Vergleich zum betrachteten Pixel sind. Zum besseren Verständnis der Konstruktion eines LBP Deskriptors siehe Abbildung 40. (Blauth, 2012)

Eine Weiterentwicklung des LBP sind die Multi-scale Block Local Binary Pattern (kurz MB-LBP). Diese sollen eine Alternative zu den Haar-like Features darstellen und unterscheiden sich vom gewöhnlichem LBP nur in der Verwendung einer weiteren Ebene mit Mittelwerten. So werden mehrere Bildregionen zu Mittelwerten zusammengefasst und auf den entstanden Block der Mittelwerte das ursprünglich beschriebenen Verfahren angewendet. Siehe hierzu Abbildung 41. (Blauth, 2012)

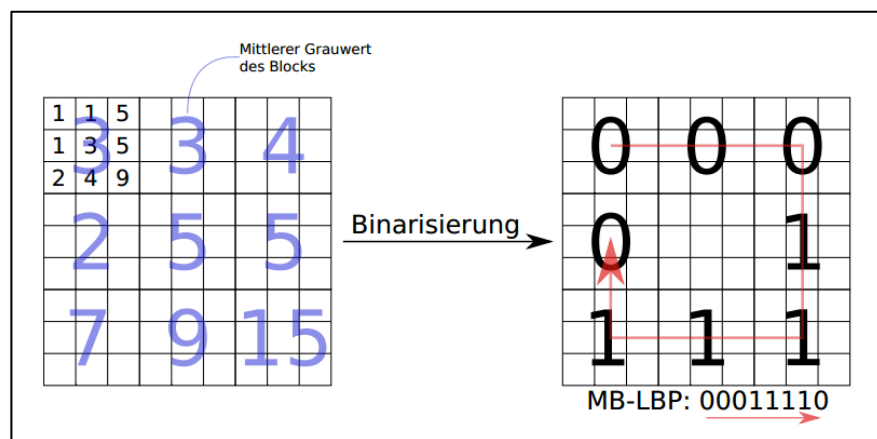


Abbildung 40 MB-LBP Konstruktion (Blauth, 2012)

Zur Vereinfachung wird nachfolgend LBP Synonym für MB-LBP verwendet.

7.1.3 Histograms of oriented Gradients (HOG)

Ursprünglich wurden die HOG-Features zur Detektion von Personen in Bildern entwickelt. Wie bei den Haar-like Features zeigte sich jedoch auch hier, dass das Verfahren zum Erkennen von anderen Objekten geeignet ist. Die Grundidee des Verfahrens ist das sich das Aussehen eines Objektes durch die Verteilung der

Helligkeitsgradienten beschreiben lässt. Hierzu werden die Gradientenbilder zu einem Eingangsbild berechnet, um mit diesen die Richtung und Magnitude des Gradienten in jedem Punkt des Bildes zu bestimmen. Anschließend wird das Eingangsbild in gleichgroße Zellen unterteilt und für jede ein Histogramm aus Gradientenrichtungen und aufsummierten Magnituden erstellt. Der Ablauf des Verfahrens ist schematisch in Abbildung 42 dargestellt. (Blauth, 2012)

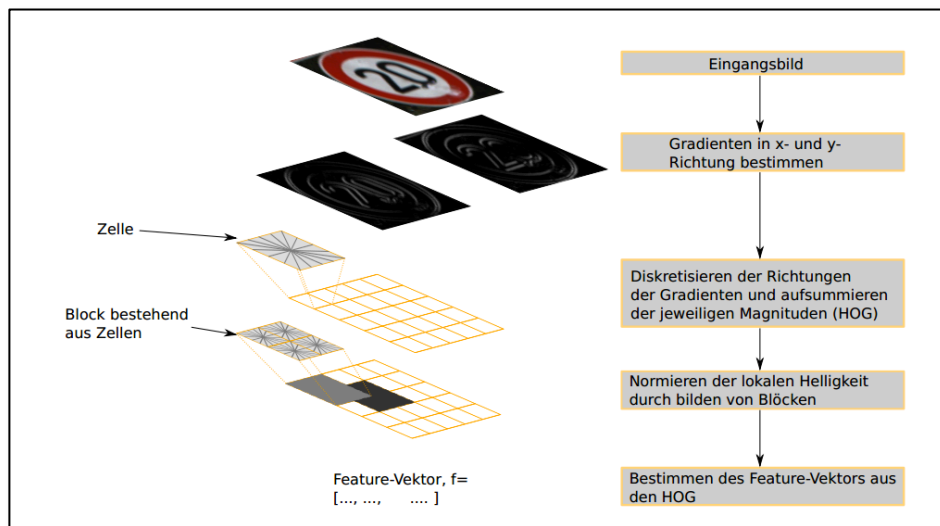


Abbildung 41 HOG-Feature Berechnungsschritte (Blauth, 2012)

7.1.4 Cascade-Classifier

Um mit Hilfe von Features ein Objekt in einem Bildausschnitt zu erkennen kann eine Kaskade von Klassifikatoren eingesetzt werden. Zur Selektion geeigneter Klassifikatoren kann der AdaBoost Algorithmus eingesetzt werden. Hierbei werden sogenannte schwache Klassifikatoren gesucht die anhand eines Merkmals ein Objekt besser als durch Raten erkennen können. Ein Klassifikator für ein Bild der Dimension x besteht dabei aus dem Merkmal f_j (Haar-like Feature bzw. LBP), einem Schwellwert θ_j (wird vom Algorithmus zur Konstruktion der Kaskade festgelegt) sowie einer Parität p_j zur Änderung des Verhältniszeichens. Die Ausgabe des Klassifikators ist durch nachfolgende Funktion bestimmt: (Blauth, 2012)

$$h_j(x) = \begin{cases} 1, & \text{wenn } p_j f_j(x) < p_j \theta_j \\ 0, & \text{sonst} \end{cases} \quad (37)$$

Nun werden solange sich ergänzende Klassifikatoren gesucht, bis ihre kombinierte Falsch-Positiv-Rate eine vorher festgelegte Schwelle f unterschreiten sowie eine Detektionsrate d erreichen. Typische werte sind hierbei $f = 0.5$ (entspricht 50%)

und $d = 0.995$ (entspricht 99,5%). Die so gefundene Menge von schwachen Klassifikatoren kann anschließend zu einem starken Klassifikator zusammengefasst werden. (Blauth, 2012)

Der Pseudocode dieses Algorithmus ist in Abbildung 43 dargestellt.

- Gegeben seien die Trainingsbilder $(x_1, y_1), \dots, (x_n, y_n)$ mit dem Label $y_i = 0$ für negative (enthalten das Objekt nicht) und $y_i = 1$ für positive Samples (enthalten das Objekt).
- Die Gewichte für $y_i = 0, 1$ werden durch $w_{1,0} = \frac{1}{2m}$ und $w_{1,1} = \frac{1}{2l}$ initialisiert. m repräsentiert die Anzahl der negativen und l die Anzahl der positiven Trainingsbilder.
- Für $t = 1, \dots, T$:
 1. Normieren der Gewichte durch $w_{t,i} \leftarrow \frac{w_{t,i}}{\sum_{j=1}^n w_{t,j}}$.
 2. Für jedes mögliche Haar-like Feature j wird ein Klassifikator h_j erstellt, welcher nur dieses Merkmal enthält. Der Fehler wird durch $\epsilon_j = \sum_i w_i |h_j(x_i) - y_i|$ ermittelt.
 3. Der Klassifikator h_t mit dem geringsten Fehler ϵ_t wird ausgewählt.
 4. Werte der Gewichte aktualisieren: $w_{t+1,i} = w_{t,i} \beta_t^{1-e_i}$ wobei $e_i = 0$ ist, falls das derzeitige Trainingsbild korrekt klassifiziert wurde, sonst gilt $e_i = 1$. Weiterhin ist $\beta_t = \frac{\epsilon_t}{1-\epsilon_t}$.
- Der endgültige starke Klassifikator ist:

$$h(x) = \begin{cases} 1, & \text{wenn } \sum_{t=1}^T \alpha_t h_t(x) \geq \frac{1}{2} \sum_{t=1}^T \alpha_t \\ 0, & \text{sonst} \end{cases}$$

mit $\alpha_t = \log \frac{1}{\beta_t}$

Abbildung 42 Pseudocode AdaBoost (Blauth, 2012)

Durch Kaskadieren der so gefundenen starken Klassifikatoren kann die Geschwindigkeit der Erkennung erheblich verbessert werden. Hierzu werden die Klassifikatoren in Stufen hintereinandergeschaltet. So kann in jeder Stufe mit dem Ergebnis „Objekt nicht gefunden“ das Verfahren ggf. frühzeitig abgebrochen werden. Siehe hierzu Abbildung 44. Die gesamte Falsch-Positiv-Rate F ergibt sich hiernach durch folgende Gleichung: (Blauth, 2012)

$$F = \prod_{i=1}^K f_i \quad (38)$$

Wobei K die Gesamtzahl der Stufen und f_i die Falsch-Positiv-Rate der i – ten Stufe angibt.

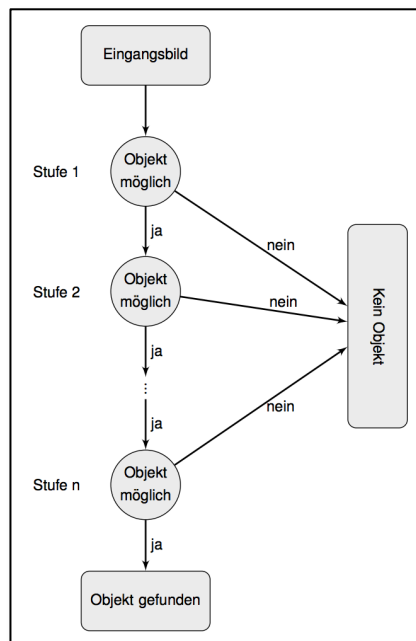


Abbildung 43 Kaskadierte Klassifikatoren (Blauth, 2012)

Die Erkennungsrate D der Kaskade ergibt sich analog mit Hilfe der Erkennungsrate der i – ten Stufe d_i durch:

$$D = \prod_{i=1}^K d_i \quad (39)$$

Für die typischen Werte $f_i = 0,5$, $d_i = 0,995$ und $K = 10$ ergeben sich demnach $F = 0,5^{10} = 9,7 + 10^{-4}$ und $D = 0,995^{10} = 0,95$. Der Pseudocode des Algorithmus zur Kaskadenkonstruktion ist in Abbildung 45 gezeigt.

- Der Nutzer wählt die maximale Falsch-Positiv-Rate pro Stufe f , die minimale Detektionsrate pro Stufe d und die maximale Falsch-Positiv-Rate der kompletten Kaskade F_{target} .
- P = Menge der Bilder, die das Objekt enthalten
- N = Menge der Bilder, die das Objekt nicht enthalten
- $F_0 = 1.0$; $D_0 = 1.0$
- $i = 0$
- **while** $F_i > F_{target}$
 - $i \leftarrow i + 1$
 - $n_i = 0$; $F_i = F_{i-1}$
 - **while** $F_i > f \cdot F_{i-1}$
 - * $n_i \leftarrow n_i + 1$
 - * Nutze P und N zum Training eines Klassifikators mit n_i Merkmalen unter Einsatz von AdaBoost [10].
 - * Prüfe derzeitige Kaskade um derzeitige Falsch-Positiv-Rate F_i und Detektionsrate D_i zu bestimmen.
 - * Verringere den Schwellwert des i -ten Klassifikators bis die Kaskade eine Detektionsrate von mindestens $d \cdot D_{i-1}$ besitzt (hat ebenfalls Einfluss auf F_i).
 - $N \leftarrow \emptyset$
 - Wenn $F_i > F_{target}$ gilt, dann prüfe die derzeitige Kaskade auf der Menge der Bilder, welche das gesuchte Objekt nicht enthalten, und füge die Falschdetektionen zur Menge N hinzu.

Abbildung 44 Kaskadenkonstruktion (Blauth, 2012)

7.1.5 Sliding Window Verfahren

Da die zuvor trainierten Klassifikator Kaskaden nur Objekte in gleicher Größe finden können ist ein weiteres Verfahren notwendig. Beim Sliding Window Verfahren wird ein Suchfenster von oben nach unten und links nach rechts über das Eingangsbild geschoben. Auf jeden so entstanden Bildausschnitt wird nun das Verfahren der Klassifikator Kaskaden angewendet und so entschieden ob in dem Fenster das gesuchte Objekt enthalten ist. Ist das Eingangsbild vollständig durchlaufen wird das Verfahren mit einem skalierten Fenster wiederholt. Dies wird solange wiederholt bis ein zuvor festgelegter Maximalwert für die Skalierung erreicht wird oder das Suchfenster größer als das Eingangsbild ist. (Blauth, 2012)

7.1.6 Training

Für das Training der Klassifikatoren mit Haar-like, HOG und LBP Features wurden 85 Bilder der Munition aus einer ähnlichen Höhe und einem ähnlichen Winkel, wie die am Roboter angebrachten Kamera aufweist, für die Objekterkennung aufgenommen. Dabei wurden die Aufnahmen auf verschiedenen Bodenbelägen, Hintergründen und Lichtverhältnissen gemacht, um eine größtmögliche Konzentration der Klassifikatoren auf die Munition zu erreichen. Für die Negativbilder wurden 700 Aufnahmen in unterschiedlichen Umgebungen und Lichtverhältnissen gemacht. Um die Munition auf den 85 Bildern für das Training einfach markieren zu können wurde ein Hilfsprogramm entwickelt. Dieses erlaubt die Angabe eines Quell- und Zielverzeichnisses und ein einfaches durchblättern der Quellbilder. Des Weiteren können mit diesem einfach Bildbereiche als Munition markiert und anschließend die Bilder zusammen mit den Koordinaten der Munition exportiert werden. Das Programm ist auf der beigefügten CD unter „source/ClassifierTraining“ zu finden. Für das Training wurde opencv mit dem AdaBoost Algorithmus und einer Fenstergröße von 43x20 Pixeln genutzt. Als minimale Detektionsrate wurde 99.9% und für die maximale Falsch-Positiv-Rate 50% gewählt. Die für das Training genutzten Bilder und durch das Training von opencv erzeugten Klassifikatoren sind auf der beigefügten CD unter „data/opencv“ zu finden.

7.1.7 Vergleich

Für den Vergleich der drei Verfahren wurde ein Testprogramm auf Basis von opencv entwickelt (auf der beigefügten CD unter „source/odroid/nerf/test“ zu finden). Dieses lädt zunächst einen angegeben zuvor trainierten Klassifikator. Anschließend werden mit der Kamera für die Objekterkennung 20 Bilder aufgenommen. Auf diese Bilder wird der Klassifikator angewendet, die gefundenen Objekte grafisch markiert und anschließend gespeichert. Dieser Test wurde für jeden Feature-Typen einmal ohne Munition und viermal mit Munition in unterschiedlichen Entfernungen und Winkeln durchgeführt. Auf den so entstanden 400 Bildern wurden anschließend manuell die korrekt und falsch markierten Bereiche gezählt und in einen Prozentsatz im Verhältnis zu den 20 Bildern umgerechnet. Die Ergebnisse dieser Tests sind in den Diagrammen in Abbildung 46 und 47 dargestellt.

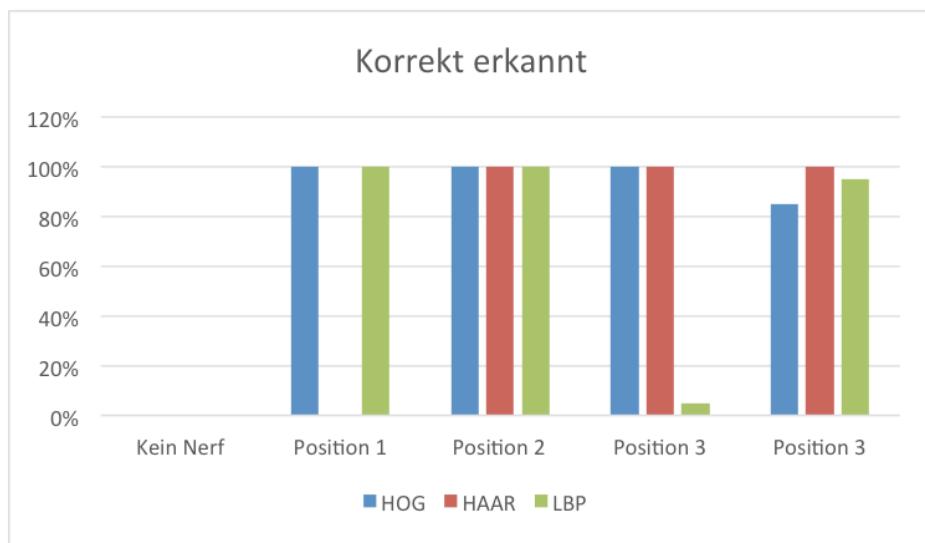


Abbildung 45 Korrekt erkannte Bereiche

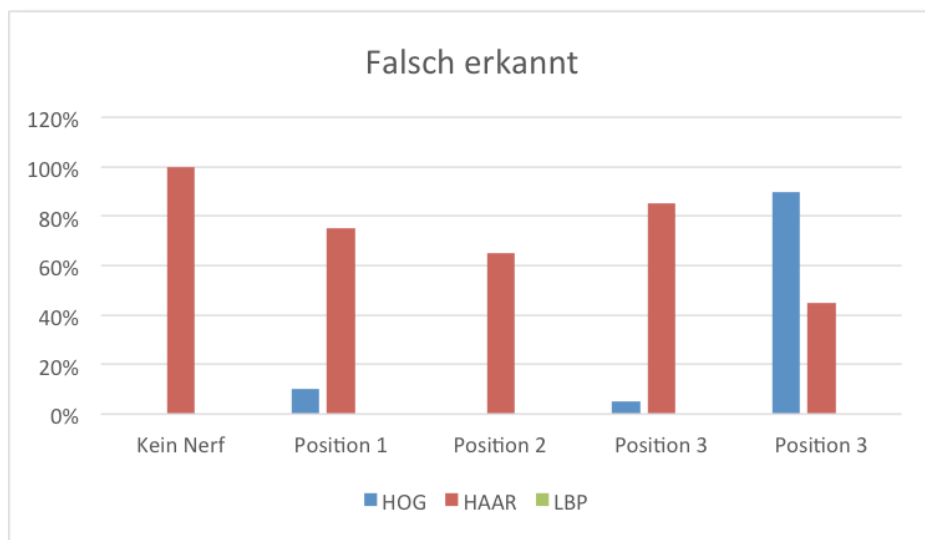


Abbildung 46 Falsch erkannte Bereiche

Hierbei ist zu sehen, dass alle Verfahren bei dem Test ohne Munition im Bild auch keine erkannte haben. Bei den Tests mit Munition im Bild zeigte sich, dass das Verfahren basierend auf LBP-Features keine Falsch-Positiv Erkennungen und bis auf einen Test auch eine sehr hohe positive Erkennungsrate aufweist. Das Verfahren basierend auf HOG-Features weist zwar eine bessere, da bei allen Tests hohe, positive Erkennungsrate als LBP auf, zeigt jedoch vor allem im vierten Test eine hohe Falsch-Positiv Erkennung. Die Haar-like Features zeigten bei den Tests die höchste Falsch-Positiv Erkennungsrate bei einer mit den LBP Features vergleichbaren positiven Erkennungsrate. In einem weiteren Test wurde auf dem Prototypen die durchschnittliche Zeit gemessen, die für die einzelnen Verfahren für

die Erkennung benötigt wurden. Wie in Abbildung 48 zu sehen schneidet das LBP-basierte Verfahren hierbei am besten ab.

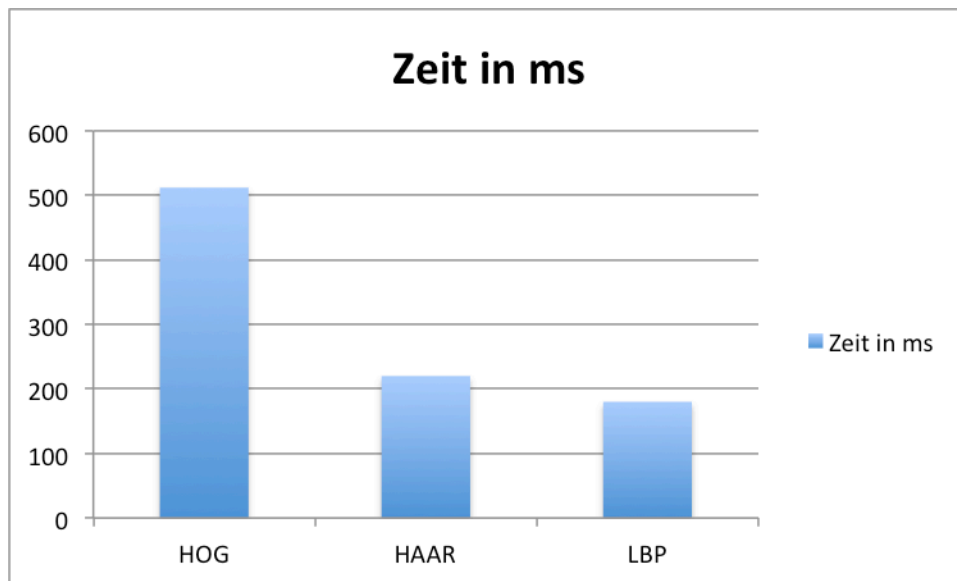


Abbildung 47 Zeitaufwand

7.1.8 Fazit

Anhand der Tests konnte gezeigt werden, dass eine auf LBP-basierte Erkennung die stabilsten Ergebnisse verspricht. So wurde bei diesen keine Falsch-Positiv-Erkennung festgestellt. Der Falsch-Positiv-Erkennung wird dabei eine größere Bedeutung zugemessen, da diese dazu führen kann, dass der Roboter versucht, andere, als Munition erkannte Objekte einzusammeln. Ein weiterer Vorteil von LBP ist, die in dem Test gezeigte Erkennungszeit. So kann mit LBP mehr als fünfmal pro Sekunde nach Munition im Bild gesucht werden, was für ein Tracking von erkannter Munition vorteilhaft ist. Es wird davon ausgegangen, dass das Ergebnis der Haar-like Features in der einfachen Form und Struktur der Munition begründet ist. Hier sind die Haar-like Features wohl vor allem aufgrund der runden Form und damit wenig passenden Features im Nachteil. Der Vorteil der Falsch-Positiv-Erkennung von LBP gegenüber HOG-basierten Features ließ sich abschließend nicht klären. Es wird davon ausgegangen, dass mit LBP mehr aussagekräftige Features für das Training der Kaskaden gewonnen werden konnten und so die bessere Falsch-Positiv-Rate zustande kommt. Als abschließendes Fazit ist festzuhalten, dass mit Hilfe der LBP-Features eine zuverlässige und leistungsstarke Erkennung der Munition möglich ist.

8 Zusammenfassung und Ausblick

8.1 Zusammenfassung

Ziel der Arbeit war, die Grundlagen für einen autonomen Roboter zu schaffen, der in der Lage ist Nerf Munition selbständig zu finden und einzusammeln. Hierzu wurde ein Prototyp entworfen der über die notwendigen Sensoren für einen autonomen Betrieb verfügt. Außerdem wurde die erforderliche Software zur Auswertung der Sensoren und Ansteuerung des Antriebs entwickelt und vorgestellt. Da der Roboter, auch nach einem Nutzereingriff, in der Lage sein soll automatisch eine Basis- und Ladestation anzufahren, wurde ein Verfahren zur absoluten Positionsbestimmung nach einem Nutzereingriff vorgestellt und untersucht. Hier konnte gezeigt werden, dass dieses Verfahren in der Lage ist nach Nutzereingriffen eine absolute Positionsbestimmung zu ermöglichen. Weiterhin wurden Verfahren zur visuellen Objekterkennung untersucht und gezeigt dass mit diesen eine zuverlässige visuelle Erkennung der Munition direkt auf der Hardware des Roboters möglich ist. So sind die Grundlagen für einen autarken und autonomen Roboter geschaffen.

8.2 Ausblick

Basierend auf dem hier Entworfenen Roboter sowie den untersuchten Verfahren zur Positionsbestimmung und Objekterkennung soll im Nachgang dieser Arbeit ein autonomes System entstehen. Hier ist ein erster Schritt die Zusammenführung der Positionsbestimmung aus Odometriedaten und dem vorgestellten Verfahren zur absoluten Positionsbestimmung. So kann ein Kalmanfilter für die Korrektur der Odometriedaten anhand der Landmarken eingesetzt werden. Die aus der absoluten Positionsbestimmung gewonnenen Entfernungen und Winkel könnten hierbei für die Gewichtung dienen. Dies verspricht eine signifikante Verbesserung der Positionsbestimmung im normalen Betrieb (ohne Nutzereingriff). Auf Basis dieser Positionsbestimmung kann anschließend ein Verfahren umgesetzt werden, dass in der Lage ist einen Raum systematisch abzufahren. So kann beispielsweise der Raum in Zellen unterteilt und mit einem Füllalgorithmus jede dieser Zellen nach Munition durchsucht werden. Eine weitere Herausforderung ist die zuverlässige Erkennung von Nutzereingriffen. Hierzu muss eine Methode gefunden werden, die in der Lage ist das Überfahren von Hindernissen von einem Nutzereingriff zu unterscheiden.

Dies könnte beispielsweise mit fahrzustandsabhängigen Schwellwerten für den Beschleunigungsmesser und für die Lage umgesetzt werden. Als letztes ist eine Mechanik zu konstruieren die in der Lage ist die Munition einzusammeln. Hierfür könnte ein Greifmechanismus und ein Auffangbehälter auf dem Roboter dienen.

Literaturverzeichnis

- Blauth. (23. 8 2012). *fraunhofer.de*. Von http://www.itwm.fraunhofer.de/fileadmin/ITWM-Media/Abteilungen/BV/Pdf/Bachelorarbeit_Marco_Blauth.pdf abgerufen
- Chang, P. H. (2012). Von <https://repository.ntu.edu.sg/handle/10356/49159> abgerufen
- Johnny Merkel, J. S. (3. 09 2014). *academia.edu*. Von http://www.academia.edu/1224626/Indoor_Navigation_Using_Accelerometer_and_Magnetometer abgerufen
- Klein. (kein Datum). *hs-coburg.de*. Abgerufen am 11. 8 2014 von http://www.hs-coburg.de/fileadmin/FB_EINF/staff/wieland/mobisys04/Positionsbestimmung_in_Gebaeuden_klein.pdf
- Ohne. (kein Datum). Abgerufen am 1. 11 2014 von <http://www.atmel.com/images/doc2543.pdf>
- Ohne. (kein Datum). Abgerufen am 19. 9 2014 von http://www.sharpsma.com/webfm_send/1205
- Ohne. (kein Datum). Abgerufen am 2. 11 2014 von <https://code.google.com/p/ratslam/>
- Ohne. (kein Datum). Abgerufen am 8. 10 2014 von Wikipedia: <http://de.wikipedia.org/wiki/Ferraris-Sensor>
- Ohne. (kein Datum). Abgerufen am 4. 10 2014 von Wikipedia: <http://de.wikipedia.org/wiki/Kreiselinstrument>
- Ohne. (kein Datum). Abgerufen am 8. 10 2014 von uni-ulm.de: https://www.uni-ulm.de/fileadmin/website_uni_ulm/iui.inst.050/vorlesungen/wise1011/les/Uebung_2.pdf
- Ohne. (kein Datum). Abgerufen am 22. 8 2014 von Wikipedia: <http://de.wikipedia.org/wiki/Absolutwertgeber>
- Ohne. (kein Datum). Abgerufen am 3. 11 2014 von Wikipedia: <http://de.wikipedia.org/wiki/HSV-Farbraum>
- Ohne. (kein Datum). Abgerufen am 13. 11 2014 von Wikipedia: <http://de.wikipedia.org/wiki/Kosinussatz>
- Ohne. (kein Datum). Abgerufen am 22. 8 2014 von Wikipedia: <http://de.wikipedia.org/wiki/Inkrementalgeber>
- Ohne. (kein Datum). Abgerufen am 28. 8 2014 von Wikipedia: <http://de.wikipedia.org/wiki/Ultraschall>
- Ohne. (kein Datum). Abgerufen am 21. 7 2014 von Wikipedia: <http://de.wikipedia.org/wiki/Beschleunigungssensor>
- Ohne. (kein Datum). Abgerufen am 7. 9 2014 von Wikipedia: <http://de.wikipedia.org/wiki/Sensordatenfusion>

- Ohne. (kein Datum). Abgerufen am 8. 10 2014 von Wikipedia: <http://de.wikipedia.org/wiki/Pulsweitenmodulation>
- Ohne. (kein Datum). Abgerufen am 12. 11 2014 von rn-wissen.de: http://rn-wissen.de/wiki/index.php/Beispiel_Drehzahlmessung_mit_Drehgeber
- Ohne. (kein Datum). Abgerufen am 25. 8 2014 von robot-electronics.co.uk: <http://www.robot-electronics.co.uk/htm/srf05tech.htm>
- Ohne. (kein Datum). (K. Publishing, Hrsg.) Abgerufen am 23. 7 2014 von klspublishing.de: <http://klspublishing.de/ejournals/e-Journ%20Av-02%20Moderne%20Gyros%20und%20AHRS.pdf>
- Ohne. (kein Datum). Abgerufen am 3. 9 2014 von Wikipedia: <http://de.wikipedia.org/wiki/Sagnac-Interferometer>
- Ohne. (kein Datum). *acroname.com*. Abgerufen am 20. 9 2014 von <http://www.acroname.com/articles/linearizing-sharp-ranger.html>
- Ohne. (kein Datum). *all-electronics.de*. Abgerufen am 12. 9 2014 von <http://www.all-electronics.de/texte/anzeigen/43860/Beschleunigungsaufnehmer>
- Ohne. (kein Datum). *altervista.org*. Abgerufen am 7. 9 2014 von <http://robbini.altervista.org/tag/complementary-filter>
- Ohne. (kein Datum). *amateurfunkbasteln.de*. Abgerufen am 21. 10 2014 von <http://www.amateurfunkbasteln.de/hbridge1/>
- Ohne. (kein Datum). *atmel.com*. Abgerufen am 1. 11 2014 von http://www.atmel.com/images/atmel-2486-8-bit-avr-microcontroller-atmega8_l_datasheet.pdf
- Ohne. (kein Datum). *chip.de*. Abgerufen am 2. 11 2014 von <http://forum.chip.de/gehaeuse-netzteile-kuehlung/pwm-unterstuetzung-1434938.html>
- Ohne. (kein Datum). *conrad.com*. Abgerufen am 3. 10 2014 von http://www.produktinfo.conrad.com/datenblaetter/175000-199999/191625-da-01-ml-RP6_DREHGEBER_SATZ_de_en.pdf
- Ohne. (kein Datum). *conrad.com*. Abgerufen am 3. 10 2014 von http://www.produktinfo.conrad.com/datenblaetter/175000-199999/191152-da-01-en-Motor_Fahrgestell_ROBBY_RP5_ROBOTER.pdf
- Ohne. (kein Datum). *conrad.com*. Abgerufen am 3. 11 2014 von http://www.produktinfo.conrad.com/datenblaetter/350000-374999/352288-an-01-en-CREATIVE_LIVE_CAM_SYNC_HD_WEBCAM.pdf
- Ohne. (kein Datum). *conrad.de*. Abgerufen am 10. 11 2014 von <http://www.conrad.de/ce/de/product/191152/Arexx-Fahrgestell-Robby-RP5RP6-Roboter?queryFromSuggest=true>
- Ohne. (kein Datum). *dexgo.com*. Abgerufen am 4. 10 2014 von <http://www.dexgo.com/index.php?site=faq/index.php&rid=2&uid=6&frage=40>
- Ohne. (kein Datum). *elcis-encoder.com*. Abgerufen am 5. 10 2014 von <http://elcis.elcis-encoder.com/TEDESCO/generalita/encincrabsocentro.html>

Ohne. (kein Datum). *elektromodellflug.de*. Abgerufen am 23. 10 2014 von <http://www.elektromodellflug.de/lipo-leitfaden-faq.html>

Ohne. (kein Datum). *homepages.thm.de/*. Von <https://homepages.thm.de/~hg7394/sns/Volumen/UltraschallSensoren.htm> abgerufen

Ohne. (kein Datum). *honeywell.com*. Abgerufen am 4. 10 2014 von http://www51.honeywell.com/aero/common/documents/myaerospacecatalog-documents/Defense_Brochures-documents/HMC5883L_3-Axis_Digital_Compass_IC.pdf

Ohne. (kein Datum). *invensense.com*. Abgerufen am 2. 10 2014 von <http://www.invensense.com/mems/gyro/documents/PS-MPU-6000A-00v3.4.pdf>

Ohne. (kein Datum). *nxp.com*. Abgerufen am 29. 10 2014 von http://www.nxp.com/documents/data_sheet/KTY81_SER.pdf

Ohne. (kein Datum). *rc-network.de*. Abgerufen am 11. 11 2014 von <http://www.rc-network.de/forum/showthread.php/43460-Erfahrung-mit-FlightPower-Hacker-EVO-20>

Ohne. (kein Datum). *reichelt.de*. Abgerufen am 1. 11 2014 von <https://cdn-reichelt.de/documents/datenblatt/A200/LM1084%23NSC.pdf>

Ohne. (2. 9 2010). *reliablecomputing.eu*. Von <http://reliablecomputing.eu/baharev-gyro.pdf> abgerufen

Ohne. (kein Datum). *robot-electronics.co.uk*. Abgerufen am 26. 8 2014 von <http://www.robot-electronics.co.uk/htm/srf04tech.htm>

Ohne. (kein Datum). *roboticlab.eu*. Abgerufen am 20. 9 2014 von http://home.roboticlab.eu/de/examples/sensor/ir_distance

Ohne. (kein Datum). *sengpielaudio.com*. Abgerufen am 20. 09 2014 von <http://www.sengpielaudio.com/Rechner-schallgeschw.htm>

Ohne. (kein Datum). *sprut.de*. Abgerufen am 29. 10 2014 von <http://www.sprut.de/electronic/temeratur/temp.htm>

Pohle, R. (kein Datum). Abgerufen am 5. 11 2014 von https://lionel.kr.hs-niederrhein.de/~pohle/gra/VL_14.pdf

Poschmann, P. (28. 9 2009). Von <http://rattlab.net/download/DA-Poschmann-Mapping-ScanMatching.pdf> abgerufen

Satoshi Suzuki, K. A. (1985). Von <http://tpf-robotica.googlecode.com/svn-history/r397/trunk/Vision/papers/SA-CVGIP.PDF> abgerufen

Wagner, C. (30. 01 2006). Von http://www.mathematik.uni-ulm.de/stochastik/lehre/ws05_06/seminar/wagner.pdf abgerufen

Abbildungsverzeichnis

Abbildung 1 inkrementall Encoder-Scheibe (Ohne, Drehgeber)	12
Abbildung 2 absolut Encoder-Scheibe	12
Abbildung 3 Signalform (Ohne, SRF05 Specs).....	15
Abbildung 4 IR-Distanzmessung (Ohne, roboticlub.eu)	16
Abbildung 5 Beschleunigungsachsen	17
Abbildung 6 Ring-Laser Gyroskop Schema (Ohne, Moderne Gyros und AHRS).....	18
Abbildung 7 Ferraris-Sensor Schema (Ohne, Ferraris-Sensor)	20
Abbildung 8 Offsetbestimmung	24
Abbildung 9 PWM (Ohne, PWM).....	24
Abbildung 10 PWM Spannung (Ohne, chip.de)	25
Abbildung 11 RP6 Chassis (Ohne, conrad.de)	27
Abbildung 12 Hardware Architektur	27
Abbildung 13 IO-Modul Anbindung.....	29
Abbildung 14 TTL-Interface (Ohne, SRF05 Specs).....	31
Abbildung 15 Spannung/Distanz (Ohne, acroname.com).....	32
Abbildung 16 Heckmodul Anbindung.....	35
Abbildung 17 Brückenschaltung (Ohne, amateurfunkbasteln.de).....	36
Abbildung 18 Entladekurve LiPo Akku (Ohne, rc-network.de).....	38
Abbildung 19 Front-/Heckmodul Ablauf	40
Abbildung 20 Ablauf Ultraschallsensoren	42
Abbildung 21 ADC-Interrupt.....	43
Abbildung 22 rtop Ausgabe	46
Abbildung 23 Aufgebauter Prototyp	48
Abbildung 24 Aufsummierender Fehler	50
Abbildung 25 Referenzpunktkorrektur.....	51
Abbildung 26 Landmarke.....	52
Abbildung 27 Sobel Eingabe (Wagner, 2006)	54
Abbildung 28 Sobel Ausgabe (Wagner, 2006)	54
Abbildung 29 Abweichung in mm	58
Abbildung 30 Entfernung zum Startpunkt.....	59
Abbildung 31 Rotationsabhängige Abweichung.....	59
Abbildung 32 Erkennungsrate	60
Abbildung 33 Nerf Munition.....	61
Abbildung 34 Nerf Munition Perspektive	61
Abbildung 35 Haar-like Features (Blauth, 2012)	61
Abbildung 36 Berechnung anhand Integralbild (Blauth, 2012).....	62
Abbildung 37 Erweiterte Features (Blauth, 2012).....	63
Abbildung 38 Integralbild geneigt um 45° (Blauth, 2012).....	64
Abbildung 39 LBP Konstruktion (Blauth, 2012).....	65

Abbildung 40 MB-LBP Konstruktion (Blauth, 2012)	65
Abbildung 41 HOG-Feature Berechnungsschritte (Blauth, 2012)	66
Abbildung 42 Pseudocode AdaBoost (Blauth, 2012)	67
Abbildung 43 Kaskadierte Klassifikatoren (Blauth, 2012)	68
Abbildung 44 Kaskadadenkonstruktion (Blauth, 2012).....	69
Abbildung 45 Korrekt erkannte Bereiche	71
Abbildung 46 Falsch erkannte Bereiche	71
Abbildung 47 Zeitaufwand	72

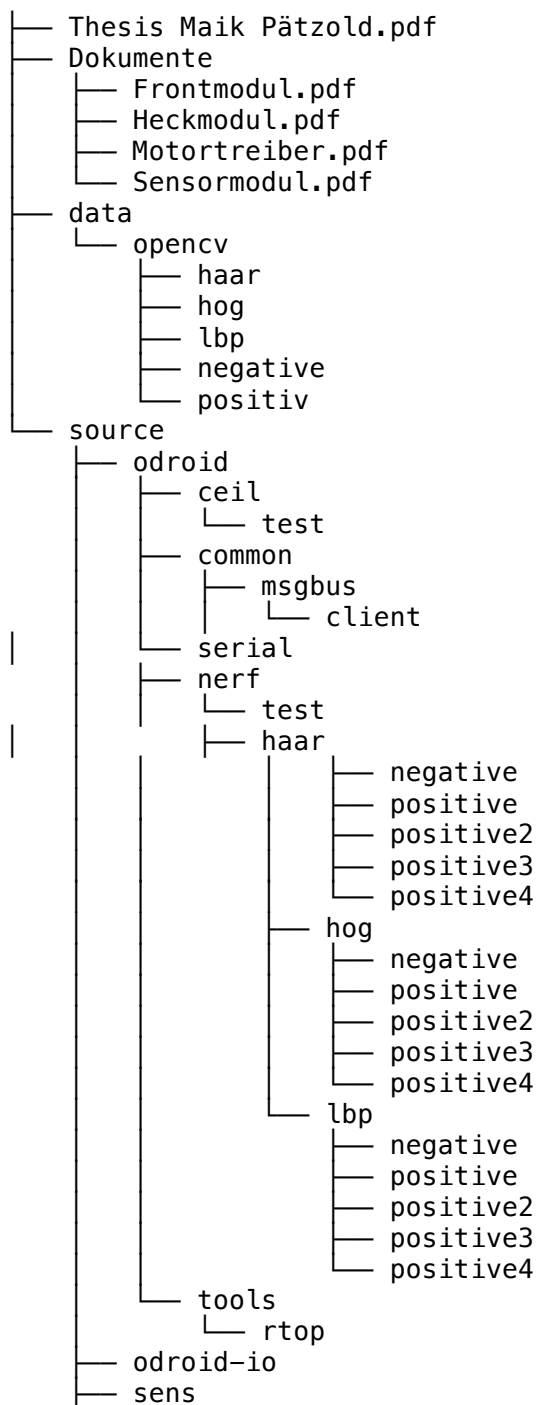
Tabellenverzeichnis

Tabelle 1 Hardwareanforderungen	28
Tabelle 2 Gegenüberstellung.....	29
Tabelle 3 Leistungsdaten	30
Tabelle 4 Leistungsdaten	32
Tabelle 5 Referenzmessung.....	33
Tabelle 6 Abschätzung der Leistungsaufnahme	39
Tabelle 7 Kostenaufstellung	48

Formelverzeichnis

Formel (1)	13
Formel (2)	13
Formel (3)	13
Formel (4)	13
Formel (5)	15
Formel (6)	15
Formel (7)	20
Formel (8)	22
Formel (9)	22
Formel (10)	22
Formel (11)	23
Formel (12)	25
Formel (13)	33
Formel (14)	33
Formel (15)	33
Formel (16)	38
Formel (17)	46
Formel (18)	53
Formel (19)	53
Formel (20)	53
Formel (21)	53
Formel (22)	53
Formel (23)	54
Formel (24)	54
Formel (25)	54
Formel (26)	56
Formel (27)	57
Formel (28)	62
Formel (29)	62
Formel (30)	62
Formel (31)	63
Formel (32)	63
Formel (33)	63
Formel (34)	63
Formel (35)	64
Formel (36)	64
Formel (37)	66
Formel (38)	67
Formel (39)	68

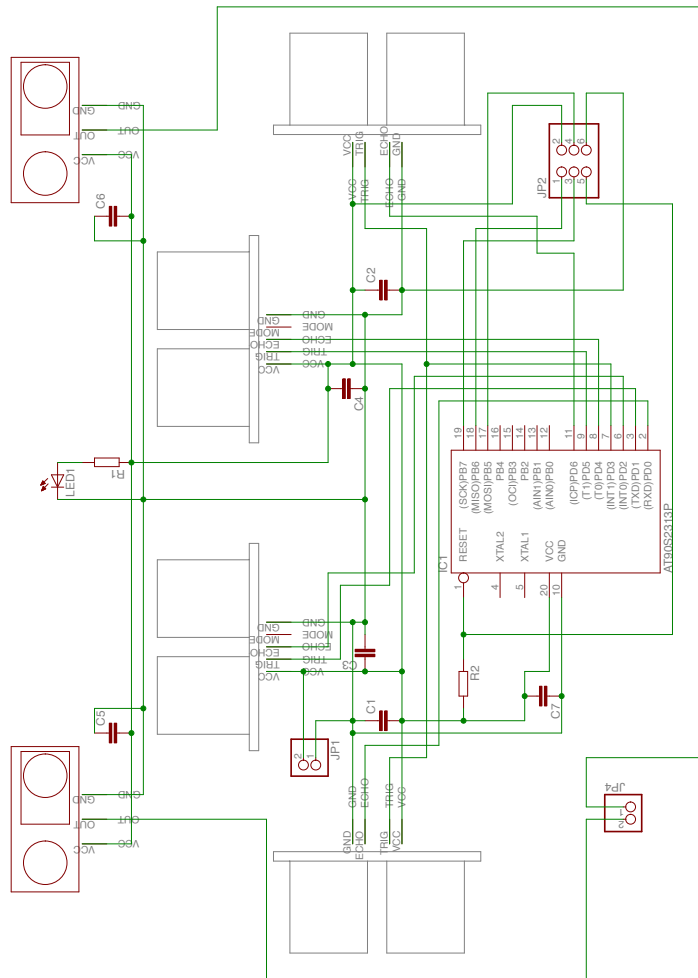
Inhalt der beigelegten CD



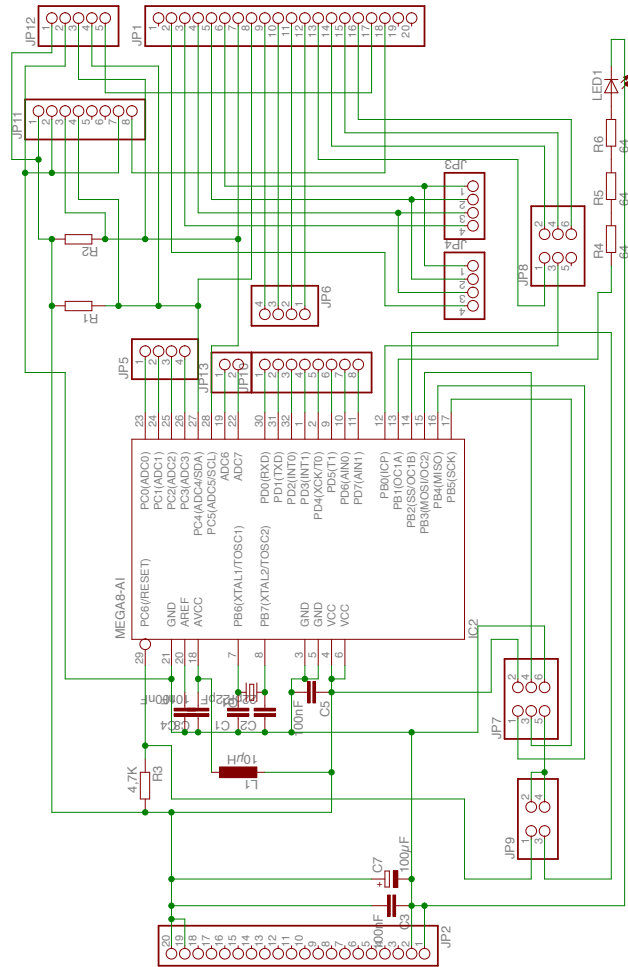
9 Anhang

- A. Schaltplan des Frontmodules**
- B. Schaltplan des Heckmodules**
- C. Schaltplan des Sensormodules**
- D. Schaltplan des Motortreibers**

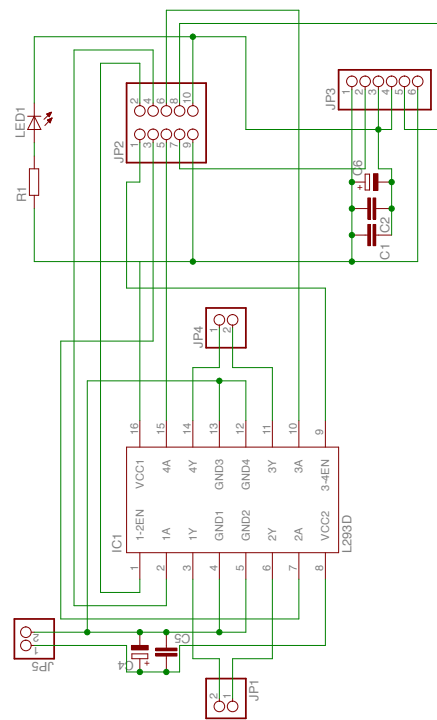
B. Schaltplan des Heckmoduls



C. Schaltplan des Sensormodules



D. Schaltplan des Motortreibers



Versicherung über Selbstständigkeit

Hiermit versichere ich, dass ich die vorliegende Arbeit ohne fremde Hilfe selbstständig verfasst und nur die angegebenen Hilfsmittel benutzt habe.

Hamburg, den _____