



Gegenstandserkennung und kategoriebasierter Transport anhand von kameraunterstützten NXT-Robotern

Studienarbeit

für die Prüfung zum
Bachelor of Engineering

von

Sebastian Hüther & Lorenzo Toso

17. März 2015

Bearbeitungszeitraum: 2 Semester

Matrikelnummer: 8853105 & 1906813

Kurs: TINF12B3

Studiengang: Informationstechnik

Ausbildungsfirma: Karlsruher Institut für Technologie

Betreuer: Gertrud Nieder

Inhaltsverzeichnis

Eidesstattliche Erklärung	IV
Abkürzungsverzeichnis	1
Abbildungsverzeichnis	2
1 Einleitung	3
2 Problemstellung	4
3 Materialien und Methoden	5
4 Hardwareumsetzung	6
4.1 Entwurf des NXT-Roboters	6
4.2 Steuerung des Roboters	6
4.3 Wahl des Kameramoduls	6
5 Softwareumsetzung	7
5.1 Wahl der Bildverarbeitungsbibliothek	7
5.2 Algorithmus zur Objekterkennung	8
5.3 Objektverfolgung	10
5.4 Entfernungsschätzung	10
5.5 Raumerkennung	12
5.6 Zielzonenerkennung	13
5.7 Hauptschleife	13

Inhaltsverzeichnis	III
6 Tests des Robotersystems	14
6.1 Tests im gesicherten Rahmen	14
6.2 Realtests	14
7 Zusammenfassung und Ausblick	15
Literaturverzeichnis	V

Eidesstattliche Erklärung

Gemäß § 5 (3) der „Studien- und Prüfungsordnung DHBW Technik“ vom 22. September 2011.
Ich habe die vorliegende Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet.

Ort, Datum

Unterschrift

Abkürzungsverzeichnis

Abbildungsverzeichnis

5.1	Beispiel einer Aufspaltung in RGB-Kanäle (links) und HSV-Kanäle (rechts)	9
5.2	Beispiel der Binarisierung des Sättigungskanals	9
5.3	Beispiel einer Segmentierung des binarisierten Sättigungskanals	10
5.4	Stereokamerabasierte Entfernungsschätzung	11
5.5	Monokamerabasierte Entfernungsschätzung	11

1 Einleitung

Im Rahmen der Studienarbeit des fünften und sechsten Semesters der Prüfung zum Bachelor of Engineering, stellt diese Arbeit eine Dokumentation zur Entwicklung eines kameragestützten Roboters dar. Ziel der Arbeit ist es mit Hilfe eines Android-Smartphones und eines LEGO Mindstorm NXT-Kits einen Roboter zu entwerfen, der Gegenstände in einem Raum erkennt, anfährt und in eine vordefinierte Zielzone transportiert. Hierfür werden diverse Methoden der Bildverarbeitung eingesetzt, welche unter der Verwendung der OpenCV-Library [1] implementiert werden.

Das nachfolgende Kapitel beschreibt die Problemstellung und erklärt eine Grundproblematik der Zusammenarbeit der beiden Hardwaremodule. Kapitel drei beinhaltet genaue Daten zu den Hardwaremodulen und deren Zusammenspiel, sowie die notwendigen Grundlagen der Bildverarbeitung, die für spätere Methoden genutzt werden. Die beiden darauf folgenden Kapitel gehen gesondert auf den genauen Aufbau des Roboters samt Konstruktionsplänen, sowie Details zur Softwareimplementierung der Objekterkennung und der Roboteransteuerung ein. Kapitel 6 beschreibt durchgeführte Tests sowohl unter speziell präparierten Bedingungen, als auch Realbedingungen. Das abschließende Kapitel ist ein letztes Fazit, welches einen Überblick über die gesamte Arbeit bildet.

2 Problemstellung

Ziel dieser Studienarbeit ist die praktische Anwendung gelernter Kenntnisse in Hard- und Software. Durch die Benutzung zweier verschiedener getrennter Module sind sowohl Kenntnisse in der Programmierung, als auch Kenntnisse in Prozessautomatisierung und in der Entwicklung verteilter Systeme erforderlich.

Konkrete Aufgabenstellung ist es einen kameragestützten Roboter zu entwickeln, der autonom Gegenstände in einem Raum erkennt und in definierte Zielzonen transportiert. Der Prozess kann hierbei in drei Teilprozesse unterteilt werden.

Erstens muss der Roboter mit Hilfe von bekannten Verfahren der Bildverarbeitung Objekte auf Grund ihrer physikalischen Beschaffenheit, beispielsweise ihrer Größe, ihrer Form und ihrer Farbe, erkennen und zielgerichtet anfahren.

Zweitens soll sich der Roboter gegenüber des Gegenstands optimal positionieren und diesen mit Hilfe eines mechanischen Greifarms aufnehmen.

Zuletzt muss der Roboter den aufgenommenen Gegenstand kategorisieren, nach der entsprechenden Zielzone suchen diese Anfahren und den Gegenstand ablegen. Die Erkennung der Zielzone kann dabei durch Markierungen an Wänden und Böden des Raumes erfolgen.

Als Bewertungskriterien dienen hierbei beispielsweise ob der Roboter alle Gegenstände erfolgreich erkennt, diese korrekt kategorisiert und in korrekte Zielzonen bewegt, sowie die Zeit in der dies geschieht.

3 Materialien und Methoden

NXT-Roboter

Android-Smartphone

Bluetooth-Verbindung

Some more Stuff

4 Hardwareumsetzung

4.1 Entwurf des NXT-Roboters

[INSERT PLANS HERE]

4.1.1 Sensoren

4.1.2 Aktorik

4.2 Steuerung des Roboters

Bluetooth Verbindung zu Smartphone

4.3 Wahl des Kameramoduls

5 Softwareumsetzung

Mit Hilfe des in Kapitel 4.3 beschriebenen Kameramoduls müssen verschiedene Aufgaben aus dem Bereich der Bildverarbeitung bewältigt werden.

5.1 Wahl der Bildverarbeitungsbibliothek

Die Umsetzung der zu bewältigenden Aufgaben kann durch die Wahl einer geeigneten Bildverarbeitungsbibliothek deutlich vereinfacht werden. Wichtige Kriterien für die Wahl der Bibliothek sind unter anderem Funktionsumfang, Dokumentation und Aktivität der Community.

5.1.1 LibCCV

LibCCV [2] ist eine open-source Bildverarbeitungsbibliothek, die viele bekannte Algorithmen implementiert. LibCCV steht unter einer BSD-Clause-3-Lizenz und kann somit für eine Studienarbeit problemlos unbegrenzt verwendet werden. Die Bibliothek ist größtenteils in C++ verfasst und somit potenziell auf einem Android-Smartphone verwendet werden. Die Verwendung auf dem Smartphone wird jedoch nicht offiziell unterstützt und kann potenziell weitere Schwierigkeiten mit sich bringen.

5.1.2 Imagemagick

Bei Imagemagick [3] handelt es sich um eine Bildverarbeitungsbibliothek, welche sehr viele Algorithmen bereits implementiert hat. Algorithmen zur Objekterkennung müssten jedoch

vollständig selbst implementiert werden, was zu einem großen zusätzlichen Aufwand führen kann. Imagemagick wird unter der Apache 2.0 Lizenz vertrieben.

5.1.3 OpenCV

OpenCV [1, 4] stellt eine der größten Open-Source-Bibliotheken für Bildverarbeitung da. Die Bibliothek hat einen starken Focus auf Echtzeitverarbeitung und wird daher auch in vielen Projekten im Bereich der Robotik verwendet. OpenCV hat eine große aktive Community, wodurch eventuelle Fragen und Probleme schnell beantwortet werden können. Zusätzlich bietet OpenCV eine offizielle Verison für Android und eignet sich somit ideal für diese Studienarbeit.

5.2 Algorithmus zur Objekterkennung

Aufgabe des Roboters, ist es Gegenstände in einem Raum mit Hilfe von Kamerabildern zu erkennen. Folglich spielt die Objekterkennung eine große Rolle.

5.2.1 Farbbasierte Objekterkennung

Einen einfachen Ansatz der Objekterkennung unter Verwendung von Methoden der in 5.1.3 beschriebenen Bibliothek OpenCV stellt eine farbbasierte Objekterkennung dar. Hierfür wird das Kamerabild zunächst vom RGB-Format [ZITAT EINFÜGEN] in das HSV-Format [ZITAT] konvertiert. Dies wird durchgeführt, da das HSV-Format unempfindlicher gegen Veränderungen in der Beleuchtung ist als das RGB-Format. Abbildung 5.1 zeigt die Aufteilung eines Bildes in die verschiedenen Kanäle.

Wie in Abbildung 5.1 zu sehen ist, eignet sich vor allem der Saturation-Kanal des Bildes um farbige Objekte zu erkennen, da dieser hohe Werte annimmt wenn die Farbintensität hoch ist.

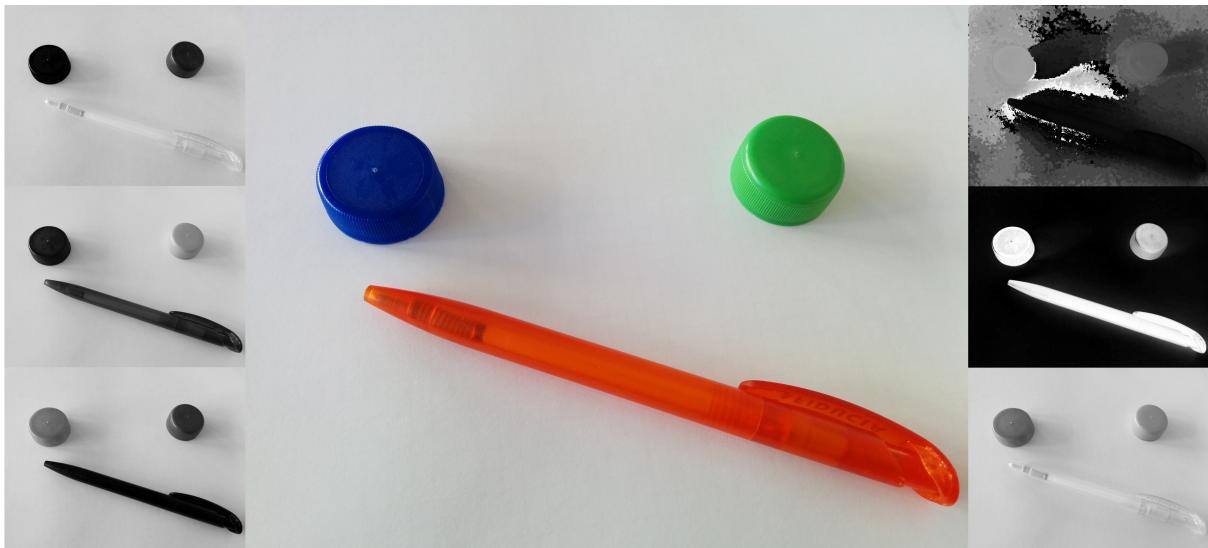


Abbildung 5.1: Beispiel einer Aufspaltung in RGB-Kanäle (links) und HSV-Kanäle (rechts)

Zuletzt erfolgt eine Binarisierung mit einem empirisch ermittelten Schwellwert von 50. Dies ist notwendig, da die, in der OpenCV-Bibliothek implementierte, Methode von Suzuki und Abe [5] zur Segmentierung von Objekten ein Binärbild erwartet. Aus dem in Abbildung 5.1 dargestellten Beispiel entsteht nach der Binarisierung schließlich Abbildung 5.2. Der Algorithmus zur Segmentierung erkennt daraus drei Objekte und zeichnet ihren Konturen, wie in Abbildung 5.3 ersichtlich, ein.

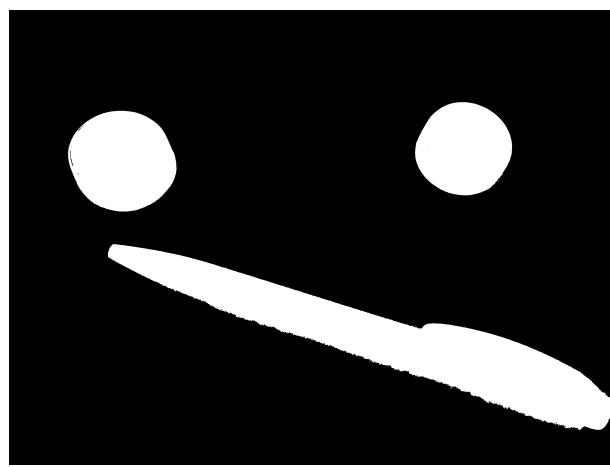


Abbildung 5.2: Beispiel der Binarisierung des Sättigungskanals

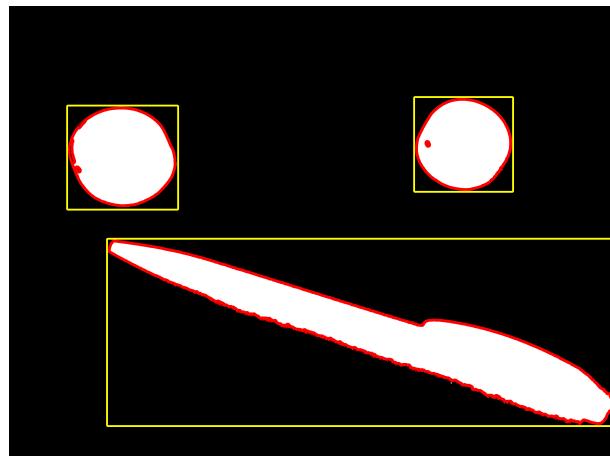


Abbildung 5.3: Beispiel einer Segmentierung des binarisierten Sättigungskanals

5.3 Objektverfolgung

5.4 Entfernungsschätzung

Die Entfernungsschätzung ist für mehrere Teile der Roboteroutine wichtig. Zunächst muss der Roboter, der ein Objekt per Kamera erkannt hat anfahren und aufheben. Hierfür muss er wissen ob der aufzuhebende Gegenstand in Reichweite des Greifarms ist. Weiterhin benötigt der Roboter die Entfernungsschätzung bei der Navigation. Er muss Hindernisse, wie beispielsweise Wände eines Raumes, erkennen bevor er kollidiert und unter Umständen seine Fracht verliert.

5.4.1 Stereokamerabasiert

Ein häufiger Ansatz für die Entfernungsschätzung stellt die Stereokamerabasierte Methode dar. Hierbei nimmt der Roboter seine Umgebung nicht mit nur einer Kamera, sondern mit mehreren, mindestens zwei, wahr. Durch einen bekannten Versatz zwischen den Aufnahmequellen der Bilder, lässt sich mittels Registrierung auch ein Versatz in den erzeugten Bildern ermitteln.

Wie die beiden menschlichen Augen, erlaubt dies das Wahrnehmen der Umgebung in dritter Dimension. Abbildung 5.4 zeigt exemplarisch eine solche Entfernungsschätzung.

Abbildung 5.4: Stereokamerabasierte Entfernungsschätzung

Der Stereokamerabasierte Ansatz ist von der Ausführung sehr elegant, da er sich an der Natur orientiert. Problematisch ist jedoch, dass zusätzliche Hardware benötigt wird. Eine weitere Kamera als bildgebendes Mittel birgt einen neuen Grad der Komplexität, da diese mit dem vorhandenen Smartphone synchronisiert werden muss. Zusätzlich müssen Rechenintensive Registrierungen vorgenommen werden, wodurch die Echtzeitfähigkeit des Systems beeinträchtigt werden könnte.

[HIER FEHLT NOCH EINE QUELLE]

5.4.2 Monokamerabasiert

Der Monokamerabasierte Ansatz stützt sich auf allen vorhandenen Informationen. Hierbei wird mittels der bekannten Position und Inklination der Kamera, sowie ihres Streuwinkels ein virtuelles Blickfeld erstellt. Dieses Blickfeld breitet sich von der Kamera aus pyramidenförmig aus und trifft den als planar angenommenen Boden. Abbildung 5.5 zeigt den Aufbau eines solchen Systems.

Abbildung 5.5: Monokamerabasierte Entfernungsschätzung

Anhand der Position des detektierten Objektes im Kamerabild lässt sich daraufhin eine ungefähre Entfernungsschätzung liefern. Problematisch ist hierbei die Ungenauigkeit des Systems. Sowohl Fehler bei der Aufnahme, als auch Probleme durch eine maximale Größenbeschränkung des detektierten Objektes können in der Praxis zu Fehleinschätzungen und damit fehlerhafter Benutzung des Greifarms führen.

[HIER FEHLT NOCH EINE QUELLE]

5.4.3 Ultraschallbasiert

Die Ultraschallbasierte Entfernungsschätzung kann auch über Große Distanzen von bis zu zwei Metern zentimetergenaue Entfernungen messen. Die Messung wird durch einen Echo-Mechanismus mit, für den Menschen unhörbaren, Schallwellen im Ultraschallbereich durchgeführt. [HIER FEHLT NOCH EINE QUELLE] Zunächst wird jedoch, ähnlich dem in 5.4.1 beschriebenen Verfahren, zusätzliche Hardware in Form eines Ultraschallsensors benötigt. Ein solcher kann jedoch direkt von LEGOTM erworben und problemlos am Roboter angebracht werden.

Das Ultraschallbasierte Verfahren kann durch eine kleine Änderung des Roboters realisiert werden und liefert sehr genaue Ergebnisse. Die Koppelung mit dem Smartphone ist bereits gegeben und genügt daher Echtzeitanforderungen.

5.5 Raumerkennung

5.5.1 Kameragestützt

5.5.2 Streckenbasiert

Über gefahrene Strecke

5.5.3 Kombination unterschiedlicher Sensordaten

5.6 Zielzonenerkennung

5.7 Hauptschleife

Arbeitszustände

1. Objekt suchen
2. Objekt ansteuern
3. Objekt aufnehmen
4. Objekt kategorisieren
5. Zielbereich suchen
6. Zielbereich ansteuern
7. Objekt ablegen

[INSERT ZUSTANDSÜBERGANGSDIAGRAMM]

6 Tests des Robotersystems

6.1 Tests im gesicherten Rahmen

6.2 Realtests

7 Zusammenfassung und Ausblick

Literaturverzeichnis

- [1] G. Bradski. Open cv. *Dr. Dobb's Journal of Software Tools*, 2000.
- [2] Libccv.
- [3] Imagemagick.
- [4] Gary Bradski and Adrian Kaehler. *Learning OpenCV: Computer vision with the OpenCV library*. Ö'Reilly Media, Inc.", 2008.
- [5] Satoshi Suzuki et al. Topological structural analysis of digitized binary images by border following. *Computer Vision, Graphics, and Image Processing*, 30(1):32–46, 1985.