



Hochschule für Angewandte
Wissenschaften Hamburg

Hamburg University of Applied Sciences

Fakultät Technik und Informatik

Department Informations- und Elektrotechnik

Bachelorprojekt

Automated Driving

RC Car Control with Open Source Image Processing

Prof. Dr.-Ing. Marc Hensel

Projektgruppe: Fabian Huber, Enzo Morino, Markus Trockel

Abgabe: DD.MM.2019

Contents

1	Einleitung	1
2	Ziel des Projekts	1
3	Kurzübersicht	1
4	Prinzip der Steuerung	1
5	Hardware	1
5.1	Raspberry Pi 3	1
5.2	Motorcontroller	2
5.3	Ultraschallsensor	2
5.4	RC Fahrzeug	2
6	Software	2
6.1	Aufbau	3
6.2	Externe Module	4
6.3	Eigene Module	4
6.4	Auswertung des Kamerabildes	4
6.5	Verwendete Repräsentationsformen von Geraden	5

1 Einleitung

2 Ziel des Projekts

Ziel des Projekts ist es, ein Modell-Auto zu bauen bzw. zu programmieren, das mithilfe eines Raspberry Pi's und einer Kamera autonom zwischen zwei Fahrbahnlinien die Spur halten kann. Das Auto soll auf einer 4 Meter langen geraden Strecke auf der Fahrbahn bleiben, einer Rechts- und einer Linkskurve mit angemessen großem Radius um 90° folgen, und eine Kreisfahrt in beide Richtungen beherrschen können. Optional soll auch eine 8-förmige Strecke wie beim Carolo-Cup befahren werden.

3 Kurzübersicht

4 Prinzip der Steuerung

Über ein Python-Programm, welches auf dem Raspberry Pi läuft, wird der Video-Stream der angeschlossenen Kamera iterativ ausgewertet. Es werden die beiden Fahrbahnlinien erkannt, durch Geraden angenähert und deren Fluchtpunkt berechnet. Auf Grundlage der x-Koordinate dieses Fluchtpunktes wird ein Ausgangssignal berechnet, welches über das PWM-Modul den Servo ansteuert, und damit den Lenkwinkel festlegt.

5 Hardware

5.1 Raspberry Pi 3

5.2 Motorcontroller

5.3 Ultraschallsensor

5.4 RC Fahrzeug

6 Software

6.1 Aufbau

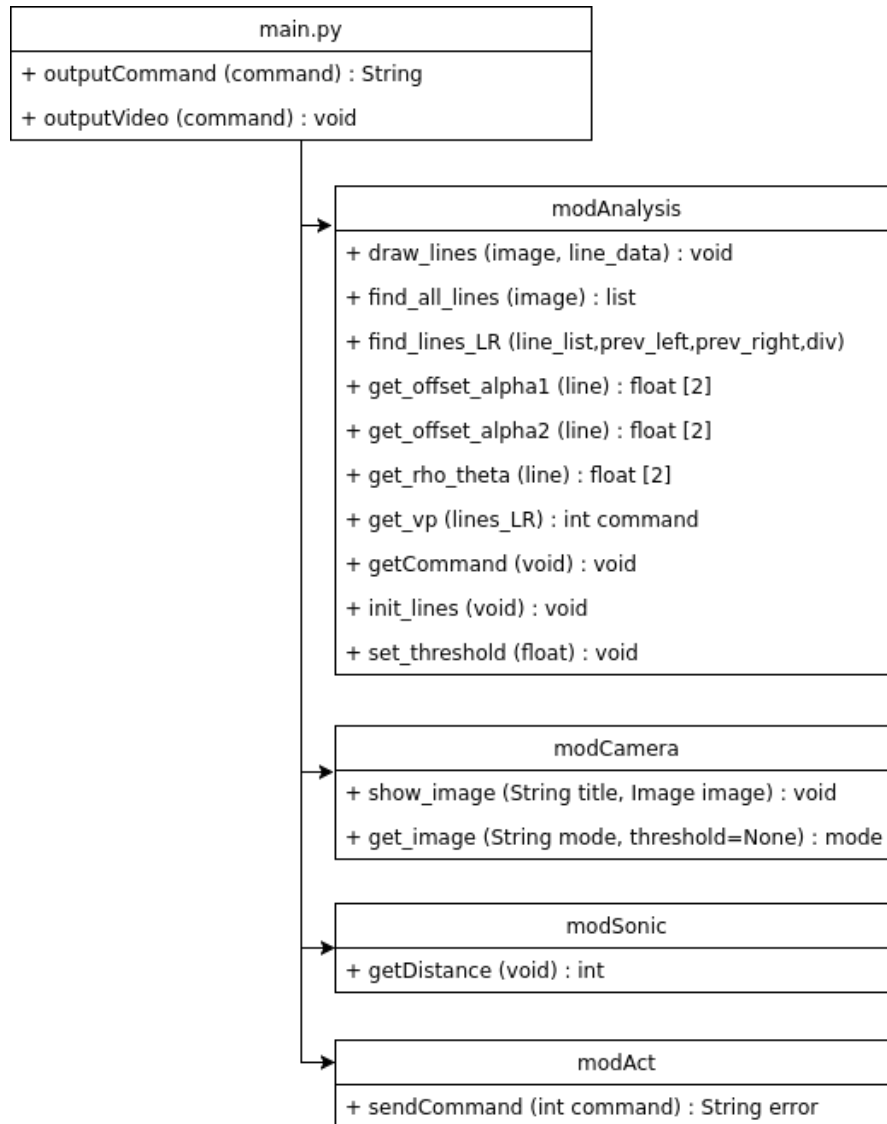


Figure 1: Aufbau des Python Codes

6.2 Externe Module

Name	Beschreibung
tkinter	...
Adafruit_PCA9685	Bibliothek zur Ansteuerung des Motorcontrollers
numpy	Bibliothek zur Verwendung von Matlab Funktionen
cv2	OpenCV 2 bietet Algorithmen zur Bildverarbeitung
io	...
time	...
importlib	...
argparse	...
pivideostream	...
picamera	...
threading	...
RPi.GPIO	Bibliothek zur Ansteuerung der GPIO ports des Raspbery Pi

Table 1: verwendete externe Python Module

6.3 Eigene Module

Name	Beschreibung
modAnalysis	Verantwortlich für die eigentliche Verarbeitung der visuellen Informationen
modAct	Verantwortlich für die Ansteuerung des Motors und der Lenkung
modCamera	Bereitet das Kamerabild für die Verarbeitung und Anzeige vor.
modSonic	Kommuniziert mit dem Ultraschallsensor und liefert Distanz zum Hindernis.

Table 2: verwendete eigene Python Module

6.4 Auswertung des Kamerabildes

Das Bild der Kamera wird iterativ ausgewertet um die beiden Fahrbahnlinien zu erkennen und daraus ein Ausgangssignal für die Steuerung zu generieren. Die Fahrbahnlinien werden durch zwei Geraden angenähert. Die beiden Geraden werden in jedem Durchlauf des Programms berechnet, indem die alle Geraden, die durch die Funktion HoughLines gefunden wurden mit dem Geradenpaar aus dem vorherigen Durchlauf verglichen werden. Durch diese vorgehensweise soll erreicht werden, dass nur die beiden Fahrbahnlinien erkannt werden und keine anderen Linien im Kamerabild.

Der Vergleich der Geraden wird durchgeführt, indem um die Geraden-Parameter aus dem letzten Durchlauf ein Toleranz-Fenster gelegt wird, und für die neuen gefundenen Geraden geschaut wird, ob sie innerhalb dieses Toleranz-Fensters liegen. Eine Schwierigkeit besteht darin die Geraden so zu repräsentieren, dass diese gut miteinander vergleichbar sind. Es kann nämlich leicht passieren, dass bei einer bestimmten Repräsentation zwei sehr ähnliche Geraden komplett verschiedene Parameter haben, und daher nicht als ähnlich erkannt werden. Daher wurden im

Programm verschiedene Repräsentationsformen verwendet, welche im folgenden beschrieben werden.

6.5 Verwendete Repräsentationsformen von Geraden

Form 1: Diese wird von der Funktion Houghlines verwendet. Sie besteht aus einem Winkel ρ und einem Abstand θ . θ ist der Winkel zwischen der Normalen, die senkrecht auf der Geraden steht, und der x-Achse. Der Winkel ρ ist die Länge dieser Normalen, d.h. der Abstand zwischen der Geraden und dem Ursprung des Koordinatensystems. Abbildung ?? zeigt ein Beispiel für die Repräsentation zweier Fahrbahnlinien mittel zweier Winkel θ_1 , θ_2 , ρ_1 und ρ_2 .

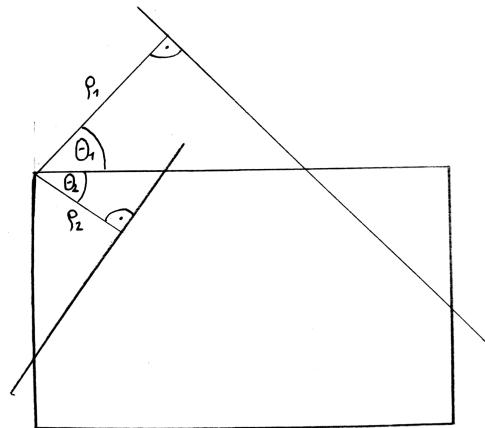


Figure 2: Beispiel für die erste Form der Geradenrepräsentation

Beim Start des Programms werden für die beiden Geraden feste Werte vorgegeben. Anhand dieser Anfangswerte werden die im nächsten Durchlauf die Fahrbahnlinien durch bessere Werte angenähert.

Dafür wird das Bild zunächst mithilfe eines Schwellwertes in Schwarzweiß umgewandelt. Bei einem geeigneten Schwellwert sollten nun möglichst nur die beiden Fahrbahnlinien weiß sein (den Wert 1 haben).

Als nächstes wird die Funktion HoughLines aus der Bibliothek cv ausgeführt, die alle Geraden im Bild sucht. Die Funktion Houghlines gibt als Rückgabewert eine Liste von Geraden zurück, die in Polarkoordinaten durch einen Winkel ρ und ein Abstand θ repräsentiert werden. ρ ist der Winkel zwischen der Normalen, die senkrecht auf der Geraden steht, und der x-Achse. θ ist die Länge dieser Normalen, d.h. der Abstand zwischen der Geraden und dem Ursprung des Koordinatensystems.