RPI-404

Pieter Haase (2052542)
Stefan Wyrembek (2065379)

ERGÄNZENDE PROJEKTDOKUMENTATION

INHALT

ZIEL DES PROJEKTS

Ziel des Projekts ist die Erweiterung sowie die Verbesserung des in dem Kurs ITS/MoSy Programmierten RPI-404. Folgende Funktionen wurden überprüft und in ihrer Funktion verbessert:

- Kalibrierung und Bewegungen der Kamera und des Radars
- Stabilisierung der Stromversorgung
- Verbesserung der Softwarearchitektur
- App Design

Des Weiteren wurde der RPI-404 um Folgende Funktionen ergänzt:

- Form- und Farberkennung mit OpenCV
- „Carry Me Home“ Funktion
- OpenCV Videostream mit Flask

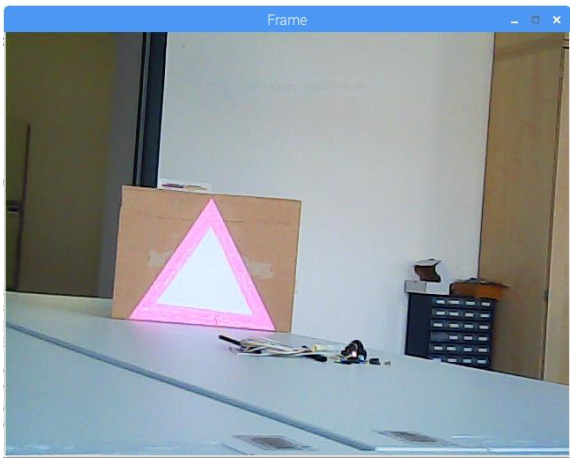
CARRY ME HOME FUNKTION

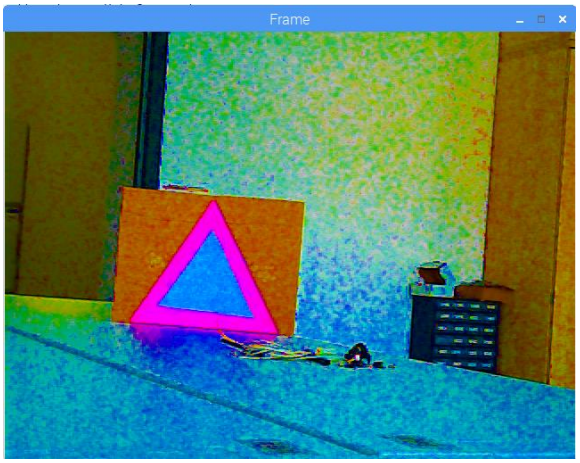
Ziel dieser Funktion ist, dass der RPi-404 selbstständig ein bestimmtes Objekt in seiner Nähe erkennt und sich autonom dessen Position annähert. Als Objekt wurde ein einfaches, pinkfarbenes Dreieck gewählt, welches auf eine Art Pappaufsteller aufgebracht wurde. Die grelle Farbe sorgt dafür, dass

das Objekt leichter von der Umgebung unterschieden werden kann und der dreieckige Umriss wird von dem Algorithmus zur Formenerkennung schnell und einfach identifiziert.

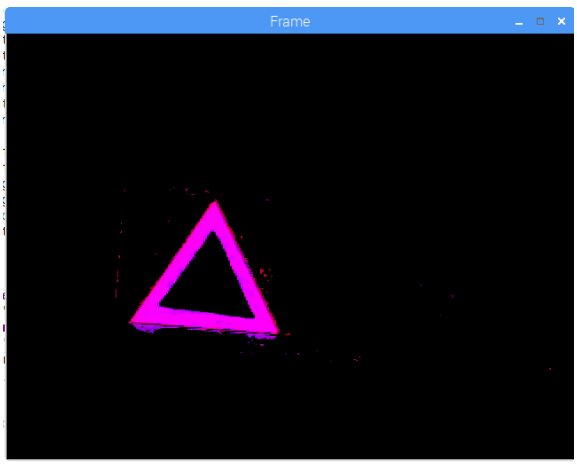
OBJEKTERKENNUNG

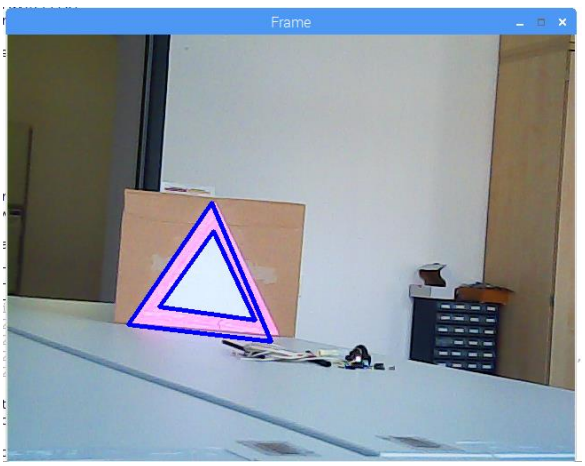
Mithilfe der OpenCV-Bibliothek für Python wird das Kamerabild der angeschlossenen Webcam verarbeitet. Zunächst wird dabei die Sättigung des Bildes erhöht, um die pinke Farbe des Dreiecks leichter erkennen zu können





Im nächsten Schritt wird von dem Algorithmus alles maskiert, was außerhalb eines Bestimmten Farb- / Helligkeits- und Sättigungsbereich ist.
Abschließend erfolgt erfolgt eine Erkennung der Umrisse innerhalb des Bildes durch einen entsprechenden Algorithmus der OpenCV-Bibliothek, sowie eine Approximation der Umrisse, bei der Umrisse mit einer Seitenanzahl von drei als Dreieck erkannt und durch ein Overlay auf dem Kamerabild entsprechend kenntlich gemacht werden.





SELBSTSTÄNDIGES FAHREN

Damit der RPi das Objekt erkennen kann, dreht er sich, sobald die Funktion aktiviert wurde, zunächst nach rechts um die eigene Achse. Dies erfolgt solange, bis das Objekt in dem Sichtfeld der Kamera auftaucht und die Dreiecksform durch den Algorithmus erkannt wird.

Befindet sich das Dreieck nun im Sichtfeld der Kamera, beginnt der RPi darauf zu zu fahren. Gleichzeitig berechnet der RPi auf welcher Seite des Bildes und wie weit entfernt von der Bildmitte sich das Dreieck befindet. Dem entsprechend leitet der RPi eine mehr oder wenige starke Lenkbewegung ein, indem die Drehgeschwindigkeit der Kurveninneren Räder reduziert wird.

Sollte der Fall eintreten, dass der RPi das Objekt aus dem Sichtfeld verliert, bleibt er stehen und dreht sich solange um seine eigene Achse, bis er das Objekt wiedergefunden hat. Dabei hängt die Drehrichtung davon ab, auf welcher Seite des Bildes sich das Dreieck zuletzt befand, als es noch im Sichtbereich des RPi war.

Ist der RPi soweit auf das Objekt zugefahren, dass es mehr als 97% der Höhe oder der Breite des Kamerabildes einnimmt, so stoppt er automatisch und überlässt dem Benutzer wieder die Kontrolle der Fahrbefehle.

ÜBERTRAGUNG DES KAMERABILDES

In der ersten Version des RPi-404 wurde das Kamerabild noch mithilfe der Motion-Bibliothek auf einer HTML-Page angezeigt, welche von der Steuerungs-App aufgerufen und dargestellt werden konnte.
Diese Methode konnten wir mit der Implementierung der „Auto Drive“-Funktion jedoch nicht mehr verwenden. Hauptgrund dafür war, dass auf das Kamerabild der Webcam durch Motion bereits zugegriffen wurde und dieses nun ohne Weiteres nicht mehr für OpenCV zur Verfügung stand. Des Weiteren wollten wir aber auch erreichen, dass die Steuerungs-App das Bild darstellt, welches von OpenCV verarbeitet wurde und in dem die Erkennung des Objektes veranschaulicht wird.

Deshalb findet die Übertragung des Kamerabildes nun mithilfe der „Flask“-Bibliothek statt. Diese bettet genau wie „Motion“ das Kamerabild in eine HTML-Website ein, nimmt dafür jedoch nicht das „Rohbild“ direkt von der Kamera, sondern das von OpenCV verarbeitete Bild.

Einziger Nachteil dieser Methode ist, dass „Flask“ in der normalen Konfiguration nur eine einzige Verbindung zu der Website erlaubt, was bedeutet, dass das Kamerabild nicht von zwei oder mehr Browsern gleichzeitig angezeigt werden kann. Mit einer entsprechenden Anpassung des Codes ließe sich dies mit Sicherheit ändern, aber da ein einziger Videostream für unsere Anwendung ausreichend war, haben wir keine weiteren Ressourcen für dahingehende Modifikationen aufgewendet.

STEUERLOGIK DER SCHRITTMOTOREN

In der ersten Version des RPi-404 haben wir festgestellt, dass die Drehung der Kamera und insbesondere die des Radars eine gewisse Ungenauigkeit aufwies. Dies führte bei der Kamera dazu, dass nach mehreren Drehbewegungen der Winkel, welcher in der App angezeigt wurde nicht mit der tatsächlichen Ausrichtung der Kamera übereinstimmte. Des Weiteren konnten wir bei der Drehbewegung des Radars feststellen, dass diese mit jedem Zyklus in eine bestimmte Richtung „gewandert“ ist, sprich der Bereich welcher von dem Radar „gescannt“ wird, langsam in eine Richtung abgedreht ist.

Bei einer Überprüfung des Codes haben wir festgestellt, dass sich ein Fehler in der Umrechnung von Motorschritten und Drehwinkel befand. Daraufhin haben wir den Code für die Drehung der Schrittmotoren kurzerhand komplett neu geschrieben und dabei nicht länger mit Drehwinkeln, sondern mit Motorschritten gearbeitet. Dies hat nicht nur den Vorteil, dass keine Fehler mehr in der Umrechnung auftauchen können, sondern bedeutet auch, dass die Drehung genauer erfolgen kann, da wir mit 512 Schritten, anstatt 360° pro Umdrehung arbeiten können.

STABILISIERUNG DER STROMVERSORGUNG

Ursprünglich wurde der Raspberry und alle Schrittmotoren über einen 4000mAh, 7 V Akkumulator betrieben. Bei Benutzung aller Motoren in Kombination mit aufwendigeren Rechenoperationen ist die Versorgung eingebrochen. Dieses Problem wurde behoben, indem eine Powerbank mit 2,4A , 5V ausschließlich für die Versorgung des Raspberrys und des vorher benutzen Akkumulators für die Motoren verwendet wurden.

APP DESIGN

Folgende Abbildungen zeigen die Grafische Entwicklung der App. Die Farbgestaltung wurde zur besseren Lesbarkeit (durch höheren Kontrast) noch einmal abgeändert. Des Weiteren wurden für die Kalibrierung von Kamera und Radar ein Button in der oberen linken Ecke eingefügt („CAL“), sowie ebenfalls ein Button, mit dem sich die „Carry Me Home“-Funktion aktivieren lässt („AUTO“). Befindet sich der RPi im autonomen Fahrmodus, so kann dieser mit erneuten druck auf den „AUTO“ Button vorzeitig gestoppt werden.