

Kooperative, verteilte Überwachung eines Gebiets auf Eindringlinge

Dennis Lisiecki, Torsten Kühl

January 5, 2015

Contents

1	Einleitung	3
1.1	Anforderungen / Lastenheft	3
1.2	Bestehende Lösungen	3
2	Technische Realisierung	5
2.1	Benötigte Module	5
3	Praktische Realisierung	6
3.1	Kommunikation der Kameras	6
4	Erklärung der Algorithmen	7
4.1	Funktionsweise der Hard- und Software	7
5	Realisierung als verteiltes System	8
5.1	Anforderungen	8
5.2	Praktische Realisierung	8
5.3	Vorteile	8

1 Einleitung

1.1 Anforderungen / Lastenheft

Ziel ist es, mithilfe von 2 (oder mehr) Raspberry-Pis, die jeweils mit Kamera, Infrarot-Sensoren, Schrittmotor und Magnetometer-Sensor bestückt sind, ein Gebiet oder einen Raum auf unerwünschte Eindringlinge zu überprüfen. Dazu werden die Geräte an unterschiedlichen Positionen aufgestellt, welche jeweils den gleichen Bereich überwachen sollen. Die Raspberries sollen zunächst ihre Umgebung auf "Artgenossen" überprüfen und diese anhand anderer Blinkfrequenzen an ihren LEDs erkennen.

Die Geräte sollen mithilfe der Kamera und eines Schrittmotoren dazu in der Lage sein, einem anvisierten Ziel automatisch zu folgen, um dieses zu beobachten und es aufzuzeichnen. Um unerwünschte Aufzeichnungen zu vermeiden (z.B. durch sich plötzlich ändernde Lichtverhältnisse), soll die Kamera mit einem PIR-Modul kooperieren. Wenn nun definitiv eine Bewegung erkannt wurde, teilt der Raspberry seine Entdeckung mitsamt der Himmelsrichtung in welcher die Bewegung entdeckt wurde, mit den anderen Raspberries, die nun in eben diese Richtung drehen um das Objekt ebenfalls zu verfolgen. Als Resultat soll an jedem Raspberry eine LED aufleuchten und ein Signal an eine App gesendet werden.

1.2 Bestehende Lösungen

Zu Beginn des Projekts haben wir als Grundlage für die Bearbeitung des Projekts das Programm Motion <http://www.lavrsen.dk/foswiki/bin/view/Motion/WebHome> als passend angesehen. Das Programm konnte nach leichter Modifikation mit der Raspberry-Pi Kamera PiNoIR verwendet werden und besaß einen funktionierenden Algorithmus zur Erkennung von Bewegungen. Außerdem stellte es einen http-Stream zur Verfügung, über welchen das aktuelle Geschehen beobachtet werden konnte und konnte automatisch Fotos und Videos aufnehmen, sobald eine Bewegung erkannt worden ist.

Als unüberwindlicher Nachteil stellte sich jedoch heraus, dass der Quellcode, welcher in C geschrieben ist, nicht so leicht vom Raspberry neu kompiliert werden konnte, wenn Änderungen daran getätigt worden sind. Das aufzeichnen von Videomaterial hat sich darüber hinaus als äußerst Ressourcenintensiv erwiesen und bleibt deswegen für unser Projekt unbrauchbar.

Nach kurzer Recherche entschieden wir uns dafür, das Projekt in Raspberry-Freundlichem

Python-Code zu verwirklichen. Aus dem Buch "Raspberry Pi - Das umfassende Handbuch" konnte Beispielcode für die Bewegungserkennung mit der PiNoIR-Kamera und dem PIR-Modul abgeschrieben werden. Diesen wollen wir um die von uns benötigten Funktionen ergänzen.

2 Technische Realisierung

2.1 Benötigte Module

- Raspberry Pi B+ *33,70 EUR*
- SD-Karte 8GB *7 EUR*
- PiNoIR Kamera *24,58 EUR*
- PIR-Modul *3,99 EUR*
- 5V-Schrittmotor *ca 7 EUR*
- Steckplatine *2,33 EUR*
- Kabel *2,01 EUR*

Somit kommt man pro Einheit auf Materialkosten von ca 82 Euro.

2.2 Verwendete Sensoren

Damit ein Computer um sich herum Bewegungen erkennen kann, benötigt es Sensoren, die in der Lage sind, die äußere Umgebung wahrzunehmen. Unser System verwendet zwei Sensoren, die jeweils auf eine völlig andere Art und Weise "sehen".

2.2.1 Raspberry Pi Kameramodul

Der erste Sensor ist eine Kamera, die wir an unseren Raspberry Pi angeschlossen haben. Zwar nimmt dieses Bewegungen nicht direkt wahr. Allerdings lassen sich die davon erzeugten Bilder so weiterverarbeiten, sodass ein Pi auch Bewegungen erkennen kann. Für unser Projekt verwenden wir das Raspberry Pi Infrarot Kamera Modul, weil es vom Raspberry Pi selbst auf jeden Fall unterstützt wird und der Support der Raspberry Pi Community hervorragend ist. Natürlich können zur Bewegungserkennung auch gewöhnliche Webcams verwendet werden. Wir geben zu bedenken, dass das Python Script, das wir zur Bewegungserkennung verwenden, nur das Raspberry Pi Kamera Modul unterstützt. Für die Raspberry Pi Kamera gibt es am Pi einen eigenen Slot, in dem das Flachbandkabel der Kamera passt.

2.2.2 PIR-Sensor

Beim zweiten Sensor handelt sich um ein Pyroelektrizität Infrarot Sensor [Zitat Raspberry Pi Das umfassende Handbuch] Hardware Der PIR-Sensor (Passive Infrared Sensor) ist einer der gängigsten Bewegungsmelder und ist oftmals auch in bewegungssensitiven Außenleuchten oder Alarmanlagen verbaut. Erkennbar ist der PIR-Sensor an seiner meist runden, milchigen Kuppel, die mit vielen einzelnen Waben versehen ist Der Sensor reagiert auf Temperaturveränderungen in seinem Sichtfeld. Somit können Menschen oder Tiere im Aktionsradius des Sensors erkannt werden. Jedoch kann der Sensor nur Veränderungen wahrnehmen. Bleibt ein Objekt ruhig im Bereich des Sensors stehen, so wird es nicht weiter erkannt. Sobald es sich weiterbewegt, schlägt der Sensor erneut an. [/Zitat] Laut Verkäufer sind die PIR Sensoren, die wir verwenden, für den menschlichen Körper gedacht. Allerdings beträgt die Abweichung der Körpertemperaturen von Säugetieren nur wenige Grad Celsius. Dementsprechend sollte der PIR auch in der Lage sein, z. B. einen Luchs zu erkennen. Um den PIR Sensor am Raspberry Pi anzuschließen, werden insgesamt 3 GPIO Anschlüssen benötigt. Einmal ein 5V Anschluss für die Stromversorgung, einen Ground und ein regulärer GPIO Pin, um den PIR zu steuern.

3 Theoretische Herangehensweise

3.1 Funktionsweise der Hard- und Software

Wenn das Programm ausgeführt wird, soll der Raspberry zunächst seine Umgebung auf andere anwesende Raspberries überprüfen und deren Koordinaten mithilfe des Magnetometers speichern. Anschließend soll die Kamera auf Bewegung achten. Sobald die Kamera eine Bewegung erkannt hat, soll das PIR-Modul prüfen, ob in der Nähe auch eine wahrnehmbare Bewegung im Infraroten Bereich erfolgt ist. Ist auch dies der Fall, reden wir von einer definitiven Bewegung. Die Himmelsrichtung wird an die anderen anwesenden Systeme geschickt, welche dann (falls nötig) ebenfalls in diese Richtung schwenken. Am Raspberry geht dann noch eine LED an, um eine erkannte Bewegung zu signalisieren.

3.2 Algorithmus für die Bewegungserkennung

3.3 Überlegungen für ein verteiltes System

4 Praktische Realisierung

4.1 Das Erkennen von Bewegungen

4.2 Überwachen eines Gebiets als verteiltes System