

Kooperative, verteilte Überwachung eines Gebiets auf Eindringlinge

Dennis Lisiecki, Torsten Kühl

I. EINLEITUNG

Tottis Baustelle

Ziel ist es, mithilfe von zwei oder mehr Raspberry Pis, die jeweils mit Kamera und Infrarot-Sensoren bestückt sind, ein Gebiet oder Raum auf unerwünschte Eindringlinge zu überprüfen.

Dazu werden die Geräte an unterschiedlichen Positionen aufgestellt, die den entsprechenden Raum aus verschiedenen Blickwinkeln zeigen. Um unerwünschte Alarmer auszuschießen, soll die Kamera außerdem mit dem Infrarot-Sensor, ("PIR-Sensor") kooperieren.

Wenn nun eine Bewegung erkannt und diese ebenfalls durch das PIR-Sensor wahrgenommen wurde, soll ein Alarm dafür sorgen, dass eine Mitteilung an das Handy des Benutzers geschickt wird und dieser entsprechend informiert wird.

Ebenfalls denkbar wäre ein Mechanismus, welcher wahrgenommene Bewegungen in ihrer Dringlichkeit beurteilt: Eine einzelne wahrgenommene Bewegung durch die Kamera eines einzelnen Raspberry Pi stellt das geringste Risiko dar. Zwei unterschiedliche Raspberry Pis, welche mit Kamera und PIR-Sensor eine Bewegung feststellen, stellen das höchste Risiko dar.

II. SYSTEM

Tottis Baustelle

A. Raspberry Pi

Der Raspberry Pi, welcher in unserem Projekt die Grundlage bildet, ist ein voll funktionsfähiger PC im Scheckkartenformat. In erster Linie wurde der Raspberry Pi mit dem Ziel entwickelt, interessierten Menschen das Erlernen von hardwarenaher Programmierung zu erleichtern. Jedes Gerät besitzt ein frei programmierbares General Purpose Input Output-Board ("GPIO-Board"). Das GPIO-Board stellt je nach Modell 26 oder 40 Pins zur Verfügung, von denen 17 bzw. 26 Pins frei programmierbar sind und die weiteren der Spannungsversorgung dienen oder als Masse dienen. Die einzelnen Pins dieses Boards lassen sich mit selbst programmierten Programmen ansteuern und können vielseitigen Zwecken dienen. So kann diverse externe Peripherie angesteuert werden, wie z.B. ein Temperatursensor, ein Ultraschall-Sensor, ein Motor oder sogar ein kleiner externer Monitor mit Touch-Funktion. Als Betriebssystem können unter anderem an die Architektur angepasste Linux-Distributionen wie *Raspbian* installiert werden. Auch Betriebssysteme, welche den Raspberry Pi

zum Mediacenter umfunktionieren, um damit Filme und Musik abzuspielen, sind von der Community mittlerweile zur Verfügung gestellt worden. Bis dato konnte sich der Raspberry Pi knapp 4 Millionen mal verkaufen und ist inzwischen in seiner vierten Version erschienen. [4] Die erste Version dieses Rechners kam Anfang 2012 auf den Markt und erfreut sich seither größter Beliebtheit. Je nach Ausführung ist das Gerät zwischen 25 und 35 Euro teuer. Die unterschiedlichen Ausführungen unterscheiden sich in gewissen Punkten:

Modell A und A+ besitzen 256 MB Arbeitsspeicher und nur einen USB-Anschluss, Modell B und B+ besitzen 512 MB Arbeitsspeicher, eine Ethernet-Schnittstelle sowie zwei, respektive vier USB-Anschlüsse. Alle Modelle müssen ohne Festplattenschnittstelle auskommen und verwenden SD-Karten als Speichermedium. Die aktuellste Version, den Raspberry Pi B+, verwenden wir für dieses Projekt.

B. Raspberry Pi Kameramodul

Für unser Projekt verwenden wir das Raspberry Pi Infrarot Kamera Modul ("Pi NoIR Camera Board"), 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. Für die Raspberry Pi Kamera gibt es am Raspberry Pi einen eigenen Slot, in dem das Flachbandkabel der Kamera passt, sodass die GPIO-Anschlüsse am Raspberry Pi vollständig für andere Aufgaben verwendet werden können. Wir geben zu bedenken, dass das Python Script, das wir zur Bewegungserkennung verwenden, nur das Raspberry Pi Kamera Modul unterstützt. Da die Kamera dazu in der Lage ist, Licht aus dem Infraroten Spektralbereich zu sehen, kann man auch nach Sonnenuntergang noch Bewegungen erkennen. Dazu muss der entsprechende Bereich mit Infrarotlicht ausgeleuchtet werden. Bei vielen Überwachungskameras wird dieses Licht mittels LEDs erzeugt, welche um das Objektiv der Kamera platziert werden. Für unser Projekt sollen allerdings vorerst keine Infrarot-LEDs zum Einsatz kommen.

C. PIR-Sensor

Beim zweiten Sensor handelt es sich um ein pyroelektrischen Infrarot Sensor:

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.[2, S. 493]

Laut Verkäufer sind die PIR Sensoren, die wir verwenden, für den menschlichen Körper gedacht.

Diesen Teil vllt damit ersetzen, dass wir den PIR erfolgreich an Tieren getestet haben?

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 3V Anschluss für die Stromversorgung, einen Ground und ein regulärer GPIO Pin, um den PIR zu steuern.

III. UNSER ANSATZ

Dennis Baustelle

Funktionsweise noch nicht final

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-Sensor 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.

IV. RELATED WORK

Dennis Baustelle

Zu Beginn des Projekts haben wir als Grundlage für die Bearbeitung des Projekts das Programm Motion[3] als passend angesehen. Das Programm konnte nach leichter Modifikation mit der Raspberry Pi Kamera verwendet werden und besaß einen funktionierenden Algorithmus zur Erkennung von Bewegungen. Außerdem stellte es einen http-Stream zur Verfügung, über welchen das aktuelle Geschehen vor der Kamera beobachtet werden und nach erkannter Bewegung automatisch Fotos und Videos aufnehmen konnte.

Als unüberwindlicher Nachteil stellte sich jedoch heraus, dass der Quellcode, welcher in C geschrieben ist, nicht so leicht vom Raspberry Pi neu kompiliert werden konnte, wenn z.B. Änderungen daran getätigt worden sind. Auch ohne Änderungen gab es Probleme bei der Kompilierung, die nach erheblichen Zeitaufwand nicht beseitigt werden konnten. Auch das aufzeichnen von Videomaterial hat sich

als äußerst Ressourcenintensiv erwiesen und bleibt deswegen für unser Projekt unbrauchbar, zumal ein Raspberry Pi nicht den nötigen Speicherplatz mitbringt, um dauerhaft aufnahmen zu machen und zu speichern.

Nach kurzer Recherche entschieden wir uns dafür, das Projekt in Ressourcenschonendem Python-Code zu verwirklichen. Im Internet konnte ein recht minimalistischer Beispielcode für die Bewegungserkennung mit der PiNoIR-Kamera gefunden werden[1]. Durch die öffentlich zugänglichen Projekte aus der Raspberry-Community konnten Lehren daraus gezogen werden, wie man einen möglichst effektiven Algorithmus schreibt, der unseren Ansprüchen genügt. Ebenfalls im Raspberry-Handbuch konnte ein Beispiel für das Arbeiten mit dem PIR-Sensor gefunden werden[2, S. 495], welchen wir in unseren Code übernehmen konnten. Diesen wollen wir um die von uns benötigten Funktionen ergänzen.

V. EVALUATION

Dennis Baustelle

VI. SUMMARY

Tottis Baustelle

REFERENCES

- [1] Oct. 2014. URL: <https://github.com/waveform80/picamera/blob/master/docs/recipes2.rst>.
- [2] Michael Kofler. *Raspberry Pi : das umfassende Handbuch ; [Grundlagen verstehen, spannende Projekte realisieren ; Schnittstellen des Pi, Schaltungsaufbau, Steuerung mit Python ; Erweiterungen für den Pi: Gert-board, PiFace, Quick2Wire u.a. in Hardware-Projekte einsetzen ; aktuell zu allen Versionen, inkl. Modell B+]*. Bonn : Galileo Press, 2014.
- [3] Lavrsen, Kenneth. *Motion - Web Home*. URL: <http://www.lavrsen.dk/foswiki/bin/view/Motion/WebHome>.
- [4] Mirko Lindner. *Über 3,8 Millionen Raspberry Pi verkauft*. Oct. 2014. URL: <http://www.pro-linux.de/news/1/21613/ueber-38-millionen-raspberry-pi-verkauft.html>.